DOCUMENT RESUME

ED 108 634 IR 002 153

AUTHOR	Wolfe, Howard
TITLE	Central Computer IMS Processing System (CIMS).
INSTITUTION	Southwest Regional Laboratory for Educational Research and Development, Los Alamitos, Calif.
REPORT NO	SWRL-TN-5-72-42
PUB DATE	30 Jun 72
NOTE	86p.
EDRS PRICE	MF-$0.76 HC-$4.43 PLUS POSTAGE
DESCRIPTORS	Attendance Records; *Computer Programs; Criterion Referenced Tests; Data Processing; Experimental Programs; Information Storage; Management Systems; On Line Systems; *Program Design; Programing; *Program Planning; Records (Forms); *Student Records
IDENTIFIERS	Computer Software; IMS; *Instructional Management System

ABSTRACT

As part of the IMS Version 3 tryout in 1971-72, software was developed to enable data submitted by IMS users to be transmitted to the central computer, which acted on the data to create IMS reports and to update the Pupil Data Base with criterion exercise and class roster information. The program logic is described, and the subroutines and functions detailed. Data specifications are included. A flowchart shows operating procedures. Relevant documents and forms are appended. (SK)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED108634

I R

SOUTHWEST REGIONAL LABORATORY
TECHNICAL NOTE

DATE: June 30, 1972

NO: TN 5-72-42

TITLE: CENTRAL COMPUTER IMS PROCESSING SYSTEM (CIMS)

AUTHOR: Howard Wolfe

ABSTRACT

This document is part of the IMS Version 3 software description. Data submitted to SWRL by IMS users is transmitted by the 690 to the Central Computer at UCLA. CIMS acts on the transmitted data to create IMS reports for transmission to the 690, and to update the Pupil Data Base with the criterion exercise and class roster information.

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

IR 000 153

This document is intended for internal staff distribution and use. Permission to reprint or quote from this work in document, wholly or in part, should be obtained from SWRL, 4665 Lampson Ave., Los Alamitos, CA.

1.0 PROGRAM IDENTIFICATION

Central computer IMS processing system (CIMS).

2.0 OBJECTIVES

2.1 To produce Pupil Performance Reports by class and unit on the criterion exercises submitted for IMS processing.

2.2 To maintain a Pupil Data Base containing the outcome scores, total scores, item responses and date of each criterion exercise processed.

2.3 To monitor the IMS data processed for consistency with the procedures the teachers were to follow. The teacher is notified of procedural problems in the Processing Conditions Report. Additionally the CIMS Run Log contains an abbreviated form of each Processing Conditions Report for SWRL's use in evaluating IMS forms and procedures.

2.4 To produce a Class Record Report confirming the roster changes that the teacher has requested.

3.0 PROGRAM DESCRIPTION

3.1 Program Logic

Using the IFACE subroutine, CIMS sorts the Data Input File and creates the Sorted Data Input File (SDIF). CIMS then processes the SDIF file by class in the following manner:

1) Locates the SDIF class to be processed in the Pupil Data Base (PDB).

2) Saves the class identification names (PDB Type 1 and 2 records) and the Class Record (PDB Type 3 records) of the class found.

3) Sets the saved Class Record status of each pupil for whom a delete request has been submitted to inactive.

4) Adds a new pupil to the saved Class Record for each add pupil request submitted.

5) Writes the saved Class Record to the new PDB on completing the processing of all requests described in steps 3 and 4 above.

6) Scores the Criterion Exercise records submitted.

7) Writes a Type 4 record containing the pupil's criterion exercise performance to the new Pupil Data Base.

8) Creates a Pupil Performance Report at the end of each criterion exercise unit. Additionally a Processing Conditions Report is generated if teacher procedural errors in administering the criterion exercise are detected.

9) Creates the Class Record Report at the end of each class if changes have been made to the Class Record.

4.0 SUBROUTINES AND FUNCTIONS

4.1 Common Blocks

Most of the arguments used by CIMS subroutines are located in common or labeled common blocks. Exhibit 1 contains a cross reference table of common blocks and subroutines which access these common blocks.

4.2 Common Block Variables

LABELED COMMON BLOCK	VARIABLE	DESCRIPTION
IBLK 2	JD(6)	Class identification of Pupil Data Base class. The elements of JD are: District number School number Teacher number Unused Grade number Class number
	ID(6)	Class identification of Sorted Data Input records. The elements of ID are the same as those in JD.
IBLK 3	JDNAME(6,5)	Class identification names (5 words/name) obtained from the Type 1 and 2 records in the Pupil Data Base. The elements of JDNAME are district name, school name, teacher name, unused, grade name, class name.
	JSITE	Site identification number obtained from the Type 1 records in the Pupil Data Base.

LABELED COMMON BLOCK	VARIABLE	DESCRIPTION
IBLK 7	NIØUT (10) or NAXTAB	Elements 1, 2, 3 ... contain the number of items/outcome for the first, second, third ... outcomes of the criterion exercise being scored.
	LEVCRI (10) or CRITAB (10)	Elements 1, 2, 3 ... contain the criterion level/outcome for the first, second, third ... outcomes of the criterion exercise being scored.
	KRITOT or TOTCRI	Unused
	ITØT or ITEMS	No. of items in criterion exercise being graded.
	NØCTØT or NØUTS	No. of outcomes in criterion exercise being processed.
	IPACK (10)	Packed outcome numbers of criterion exercise being processed e.g., if a criterion exercise is made up of outcomes 2 and 4 IPACK (1) = 2 and IPACK (2) = 4.

LABELED COMMON BLOCK	VARIABLE	DESCRIPTION
		NOTE: All values in IBLK7 are derived from the appropriate criterion exercise record in the Test Descriptor File.
IBLK 11	IERR (2,100)	First Subscript is 1 for the error code number and 2 for the pupil/page number that is in error. Second subscript allows up to 100 error messages/class.
	MREL	MREL counts the number of error codes set in IERR.
IBLK 12	ITYPE	This variable notes the record type of the last Pupil Data Base record read or written. If the last operation was a write the ITYPE is set equal to the record type multiplied by -1.
IBLK 20	JYEAR, JMONTH, JDAY	Year, month, and day the Data Input tape was created.
	JDENT	Data Input Tape number.
	JTIME, JOVER	Unused
IBLK 21	IYEAR IMONTH IDAY	Year, month and day criterion exercise taken.

LABELED COMMON BLOCK	VARIABLE	DESCRIPTION
IBLK 22	JHI	Highest pupil number in a given Class Record.
IBLK 23	ISWRL	Set to logical unit number that reports are to be output to.
IBLK 24	MSITE	Site identification number of the last class processed.
JBLK 3	IRESP (28)	This array contains the item responses made on one criterion exercise.
	IUNIT	Criterion exercise unit number.
	IPROG	Criterion exercise program number.
	IPOS	Pupil identification number.
JBLK 4	ISCORE (60, 10)	ISCORE array contains up to 10 scores (9 outcomes and total) by pupil identification number (1-60).
JBLK 7	IPAGE	Unused
	LINES	Line count of report page being processed.
JBLK 25	LPROG	Program number of Test Descriptor record currently loaded.
	LUNIT	Criterion exercise unit number of Test Descriptor record currently loaded.

LABELED COMMON BLOCK	VARIABLE	DESCRIPTION
	LUNIT	Criterion exercise unit number of Test Descriptor record currently loaded.
	KEYBG	KEYBG contains the initial word position of the answer key items of the current Test Descriptor record.
	ITMBG	ITMBG contains the initial word position of the no. of items/ outcome string in the curent Test Descriptor record.
	KRIBG	KRIBG contains the intial word position of the criterion level/ outcome string in the current Test Descriptor record.
	ITOTBG	ITOTBG contains the word position of the number of criterion exercise items in the current Test Descriptor record.
	KRIBG	KRIBG contains the word position of the total score criterion level in the current Test Descriptor record.

LABELED COMMON BLOCK	VARIABLE	DESCRIPTION
	KEY (50)	The KEY array contains the answer key items of the criterion exercises being processed. The correct response, for item 1 is in KEY (1), for item 2 is in KEY (2),... The answer keys are length one alphanumeric characters.
	NOUT (50)	Array NØUT contains the outcome assignment of each criterion exercise item.
JBLK 28	JEBUF (43)	This array saves the last Type 4 or 5 record read from the Pupil Data Base.
JBLK 29	ICODE	ICODE = 100* (Criterion Exercise unit no.) + Pupil Identification no. where the unit and pupil number are from the Data Input File record being processed.
	JCODE	JCODE = 100* (Criterion Exercise unit no.) + Pupil Identification no. The unit and pupil number are from the Type 4 Pupil Data Base record last read by CIMS.

LABELED COMMON BLOCK	VARIABLE	DESCRIPTION
JBLK 30	IEBUF (13,60)	This array contains the Class Record data of the class being processed.
KBLK 1	IADDR (500, 6)	This array saves up to 500 Data Input file sort keys. Each of these keys contain 7 elements namely: 1) District/School/Teacher identification numbers 2) Grade/Class identification number 3) Unused 4) Criterion exercise unit number 5) Pupil identification number 6) Page number
KBLK 2	ISIG (60)	This array contains the pupil identification numbers of those pupils in a class for which criterion exercise data was submitted. In the UPDATE subroutine each ISIG entry is prefixed with group number.
	JSIG (60)	This array contains a list of pupils to be deleted from a given Class Record.

LABELED COMMON BLOCK	VARIABLE	DESCRIPTION
		generated. Any ensuing Data Input File records with the same ID as that in KD will have the system message suppressed.
(UNLABELED COMMON)	K4	K4 contains the position number of the Tested Descriptors record to be accessed.

4.3 Subprogram Descriptions

SUBPROGRAM	DESCRIPTION
COMP	This integer function is used as an external function by SORT to sort the DATA INPUT file. The arguments are: 1) Position of 1st record to be compared. 2) Position of 2nd record to be compared. These pairs of indices are provided by SORT. COMP compares the record keys saved in IADDR. Each key consists of 5 elements: 1) district/school/teacher id. numbers ('/' means concatenated into 1 word) 2) grade/class id. numbers

SUBPROGRAM	DESCRIPTION
	3) criterion exercise unit number 4) pupil number 5) page or list number COMP returns a "-1" if the keys are out of sequence, a "0" if the keys are equal, and a "1" if the keys are in sequence. Additionally, if two record keys are identical, COMP flags this event by storing a negative integer in the first word of each of the involved record pairs. These flags are not set if the program number is invalid.
COMP 1	This integer function is used as an external function by SORT. The arguments are the same as those of COMP. COMP 1 compares the concatenated group/pupil number values in the ISIG (See KBLK 2 in Section 4.2) array to sort each class by pupil number within group.
COMP 4	This integer function is used as an external subroutine by SORT. The arguments are the same as those of COMP. COMP 4 compares the error message codes

SUBPROGRAM	DESCRIPTION
	and pupil/page numbers in the array IERR to sequence this array by pupil/page number within error code.
ERR	The ERR subroutine prints system error messages in the CIMS Run Log, and saves Processing Condition Report error codes in array IERR (See IBLK 11 in Section 4.2). Additionally, a list of each message code entered in IERR is output to the CIMS Run Log. The subroutine arguments are: 1) The return statement number in the calling subroutine or program 2) Error code number 3) Page/Pupil Code/NIL depending on error code used. 4) Unused 5) Unused 6) Unused The error code numbers are positive for Processing Conditions messages and negative for system error messages.
ERROR	Subroutine ERROR sorts array IERR using COMP 4, then uses the sorted IERR array to print the Processing Conditions Report.

SUBPROGRAM	DESCRIPTION
FIND	Subroutine FIND locates the class to be processed in the Pupil Data Base. FIND reads/writes the Class Identification names records (Types 1 & 2) then reads and saves the Type 3 (Class Record) records.
IFACE	IFACE reads the Data input file, transfers the Class Identification and date information of the Class ID Sheet to all following records for that class. All records except the Class Identification Sheets are saved in array IBUFF (See KBLK 9 in Section 4.2). The record keys are saved in array IADDR (See KBLK 1 in Section 4.2). IBUFF is then sorted using SORT and COMP. The sorted records are then written to the SORTED DATA INPUT file. Delete and add pupil requests are forced to sort at the front of each set of criterion exercises records of a class by setting the criterion exercise field negative.
INPUT	Entry point INPUT in subroutine IOIBM is used by CIMS to perform all read operations. For sequential files the arguments used are:

SUBPROGRAM

DESCRIPTION

1) File identification number
2) Record type number
3) Input buffer (array)
4) Length of the input buffer in words

For the Test Descriptor random access file the arguments are:

1) 3
2) 1 - Position disk
   2 - Read disk
   3 - "Rewind" disk
3) Input buffer word number 1 contains the logical unit number of the Test Descriptor File. Input buffer word number 2 contains the sequence number of record to be read.
4) Two for position disk operation, otherwise one.

The file identification numbers are:

2 - Pupil Data Base
3 - Test Descriptor Data Base
4 - Data Input File
5 - Sorted Data Input File

SUBPROGRAM	DESCRIPTION
IØIBM	See INPUT and OUTPUT.
LDR	See LDRIN and LDROUT.
LDRIN	Entry point LDRIN in subroutine LDR reads the Type 1 and 2 records in the Pupil Data Base and saves the class identification names, the remote site identification number and the class identification numbers contained in these records.
LDRØUT	Entry point LDRØUT in subroutine LDR writes Type 1 and 2 records to the Pupil Data Base using the information stored by LDRIN.
LIZ	See LIZ1, LIZ2, and LIZ3.
LIZ1	LIZ1 is an entry point in subroutine LIZ. LIZ1 provides a tally of the number of pupils/pages that have a given error code. This tally is then included in the Processing Conditions Report message produced in the calling subroutine (ERROR). The arguments are: 1) The KEY array which contains the sorted sequence of the error codes in array IERR (See IBLK 11 in Section 4.2). 2) The index which references the point

PROGRAM	DESCRIPTION
	in IERR at which LIZ1 is to begin. On completion the index points to the next error code in array IERR. 3) LIZ1 tally of the number of pupils/ pages which had the same error code. 4) This value contains the error code number to be processed by LIZ1.
LIZ2	LIZ2 is an entry point in subroutine LIZ. LIZ2 lists all pupils with the same error code in the Processing Conditions Report. If the fourth argument is equal to -1 only the pupil numbers are listed, otherwise both the pupil number and name are listed. The arguments are: 1) The return statement number in the calling subroutine (ERROR). 2)tthrough 5) are identical to 1) through 4) of LIZ1.
LIZ3	LIZ3 tallies the number of times a given pupil or page received the same error code. This tally is included in the appropriate Processing Condition Report message. The arguments are the same as those in LIZ1.
MAIN	The MAIN program drives the other subroutines used in CIMS. Additionally MAIN, 1) Processes add pupil requests

SUBPROGRAM	DESCRIPTION
	2) Processes delete pupil requests 3) Stores the item responses on the criterion exercise being processed in array ISCØRE. 4) Performs most of the error checks in CIMS.
NAME	See NAMIN and NAMØUT.
NAMIN	Entry point NAMIN in subroutine NAME reads in the Type 3 (Class Record) records from the Pupil Data Base and stores this information in array IEBUF (See JBLK 30 in Section 4.2)
NAMØUT	Entry point NAMEØUT in subroutine NAME writes the Class Record (IEBUF array) to the new Pupil Data Base.
NTRAN	The NTRAN subroutine is used to perform I/Ø requests on random access files. Used on the 360/91 at UCLA NTRAN simulates the UNIVAC 1108 library routine "NTRAN." In case of CIMS, NTRAN is used to access the Test Descriptor file. The arguments are: 1) Logical unit number 2) Operation code: 1 - WRITE 2 - READ

SUBPROGRAM	DESCRIPTION
	6 - POSITION 10 - REWIND 22 - NØP
OUTPUT	Entry point ØUTPUT in subroutine IØIBM is used by CIMS to perform all write operations except for printer output. The arguments used are: 1) File identification number 2) Record type number 3) Output buffer (array) 4) Length of the output buffer in words. The file identification numbers are: 2 - Pupil Data Base 4 - Sorted Data Input File
PAGE	This subroutine prints the heading lines of the Pupil Performance, Class Record and Processing Conditions Reports. The arguments are: 1) Criterion exercise unit number (not used in the Class Record Report). 2) 99 - indicates that the Class Record Report headings are to be printed. 98 - indicates that the Processing Conditions Report headings are to be printed. If the argument is neither 98 or 99 the Pupil Performance Report headings are to be printed.

SUBPROGRAM	DESCRIPTIONS
REP	Subroutine REP prints the Pupil Performance Report.
SCORE	The SCORE subroutine scores criterion exercise records by outcome and total. This is accomplished by retrieving the answer key and outcome assignments from the Test Descriptor File.
SØRT	This integer function is a general purpose in-core sorting routine. SORT was written by W.V. Snyder of the Jet Propulsion Laboratory. The routine compares the record keys of record pairs in a user written external subroutine (See argument 4 below). The arguments are: 1) The KEY array which contains the index number of each record in sorted order on completion of SORT. A successor technique is used, i.e., to get the next record set I = KEY (I). The initial value of I is returned by the SORT function. 2) The IHEAD array which must be equal in size to the KEY array and is used as a work area.

SUBPROGRAM	DESCRIPTION
	3) The value of this argument indicates the number of records to be sorted. 4) This argument contains the name of the user written external subroutine. See COMP, COMP1 and COMP4 which are used to perform different sorts in CIMS.
SPACE	In creating the print images of the Pupil Performance, Processing Conditions or Class Record reports a page can be "ejected" by calling this subroutine which calculates the number of remaining lines for the page and prints this information in the form of a 690 control record. For example, if 10 lines are needed to complete the page the 690 control record printed would be '==10'.
STATUS	This subroutine is used to print the Class Record Report.
TOP	TOP prints the appropriate Class Record messages. The arguments are: 1) The value 1 causes 'The following pupils have been deleted from the class record' message to be printed. The value 2 causes 'The following pupils

SUBPROGRAM	DESCRIPTION
	have been added to the class record'
	have been added to the class record' message to be printed.
	2) Unused.
	3) Unused.
UPDATE	The UPDATE subroutine inserts Type 4 records into the new Pupil Data Base. Each record contains the information retained for one criterion exercise.

5.0 DATA SPECIFICATIONS

5.1 Data Input File

The Data Input File contains 3 different record types. However, only one format is needed because the fields of each record type are staggered on the 80 position record. The record types are:

1 - Class Identification

2 - Pupil Update

3 - Criterion Exercise

Record layout is found in Exhibit 2.

5.2 Sorted Data Input File

Subroutine IFACE processes the Data Input File to create sorted Data Input File. The format of the Sorted Data Input is found in Exhibit 3.

5.3 IMS Test Descriptor File

Purpose

The Test Descriptor File contains all information required to score the multiple-choice criterion exercises used in IMS. Since the file is located on drum storage, the answer key and others related scoring information (e.g., items per outcome, criterion level per outcome, etc.) within the file are readily accessible for scoring. The Test Descriptor File is designed for direct access using relative record number which is a function of instructional program number and unit number.

File Structure

As shown below, the file is partitioned into blocks, each of which corresponds to a SWRL program. Each record within a program block corresponds to a specific instructional unit within that SWRL program.

Program 1
(e.g., Instructional Concepts Program)

- Test Descriptor Record for the first Criterion Exercise (pad each record to the size of the largest record in file)
- Test Descriptor Record for the second Criterion Exercise
- ⋮
- Test Descriptor Record for the last Criterion Exercise of Program 1
- Pad to size of largest program block

Program 2

- Test Descriptor Record for the first Criterion Exercise (pad each record to the size of the largest record in the file)
- Test Descriptor Record for the second Criterion Exercise
- ⋮
- Test Descriptor Record for the last Criterion Exercise of Program 2
- Pad to size of the largest program block

Program N

- Test Descriptor Record for the first Criterion Exercise (pad each record to the size of the largest record in the file)
- Test Descriptor Record for the second Criterion Exercise
- ⋮
- Test Description Record for the last Criterion Exercise of Program N
- Pad to size of largest program block

Record Structure

All fields contain integer values, except for the answer key field, whose contents will be alphnumeric. The record format is described in Exhibit 4.

5.4 IMS Pupil Data Base

The Pupil Data Base is structured by class. Each class contains

a set of Type 1, 2, 3, 4 and 5 records. All records contain the class identification number which is a concatenation of district, school, teacher, grade and class numbers. Classes are sequenced by class id. no. Within class the sequencing is by record type number. Type 3 records are sequenced by pupil number. Type 4 records are sequenced by criterion exercise unit and pupil numbers.

Each run of CIMS merges the Data Input File with the old Pupil Data Base and creates a new Pupil Data Base.

Record Type 1: Class Identification Record

The Class Identification Record contains the record type number, the district name, the school name, the teacher name, the site number and the class identification number. See exhibit 5.

Record Type 2: Continuation of the Class Identification Record

The Continuation of the Class Identification Record contains the record type number, the program name, the grade level name, the class name and the class identification number. See Exhibit 6.

Record Type 3: Pupil Name Record

The Pupil Name Record contains the record type number, the student's last, first and middle names, the class identification number, and the pupil number. See Exhibit 7.

Record Type 4: Pupil Scores Record

The Pupil Scores Record contains one pupil's criterion exercise scores by outcome and total score. In addition, the pupil's individual responses are retained together with the data identifying the record (date, unit, group, class identification and pupil numbers). See Exhibit 8.

Record Type 5: Class End Delimiter Record

The Type 5 Record is used either to mark the end of one class' records. This record is similar to a Type 4 record except that the unit and pupil number are set = 99 and the data fields set = 0.

File End Delimiter Records

Type 1 and type 2 records with the class identification fields set to 999...end the Pupil Data Base.

5.5 Pupil Performance Report

This report lists the percentage outcome scores achieved by the pupils in a given class on a given criterion exercise. A sample Pupil Performance Report is found in Exhibit 9.

5.6 Processing Conditions Report

Procedural problems on the part of the teacher or remote scanner operator are noted in the Processing Conditions Report. A complete list of messages is found in Exhibit 10. Message number 1020 in this exhibit is used as a heading to messages

numbers 1021-1025. The error checking sequence performed by CIMS is found in Exhibit 11. A sample Processing Conditions Report is found in Exhibit 12.

5.7 Class Record Report

This report lists the changes that have been made to the Class Record. A sample Class Record Report is found in Exhibit 13.

5.8 CIMS Run Log

This log lists the following:

1) Run date
2) Time run submitted
3) Run id. (Data Input File Tape number)
4) No. of cover sheets (Class id. records in Data Input File)
5) No. of data sheets (Non-Class id. records in Data Input File)
6) The Type 1 and Type 2 records of each class in the Pupil Data Base
7) The Class Record (Type 3 records) of each class processed in the CIMS run.
8) No. of additions and deletions made to each Class Record.
9) A summary by class and criterion exercise unit of the number of criterion exercises processed.
10) The message number of each Processing Conditions message printed in the Processing Conditions Report.
11) The system messages detected in the CIMS run.
12) The number of old Pupil Data Base records.
13) The number of Type 3 records added to the Pupil Data Base.

14) The number of Type 4 records added to the Pupil Data Base.

15) The total number of records in the new Pupil Data Base.

The system messages are found in Exhibit 14.

6.0 PROGRAM CONSTRAINTS AND LIMITATIONS

6.1 Programming Language

Fortran IV

6.2 Vendor

Campus Computing Network at UCLA

6.3 Storage Requirements

Region Size = 250K

6.4 Hardware Configuration

360/91 computer, 2 tape drives and disk.

6.5 Program Parameters

Maximum of 500 Data Input File records can be processed in 1 CIMS run.

6.6 Error Messages

See Exhibit 14.

7.0 OPERATING INSTRUCTIONS

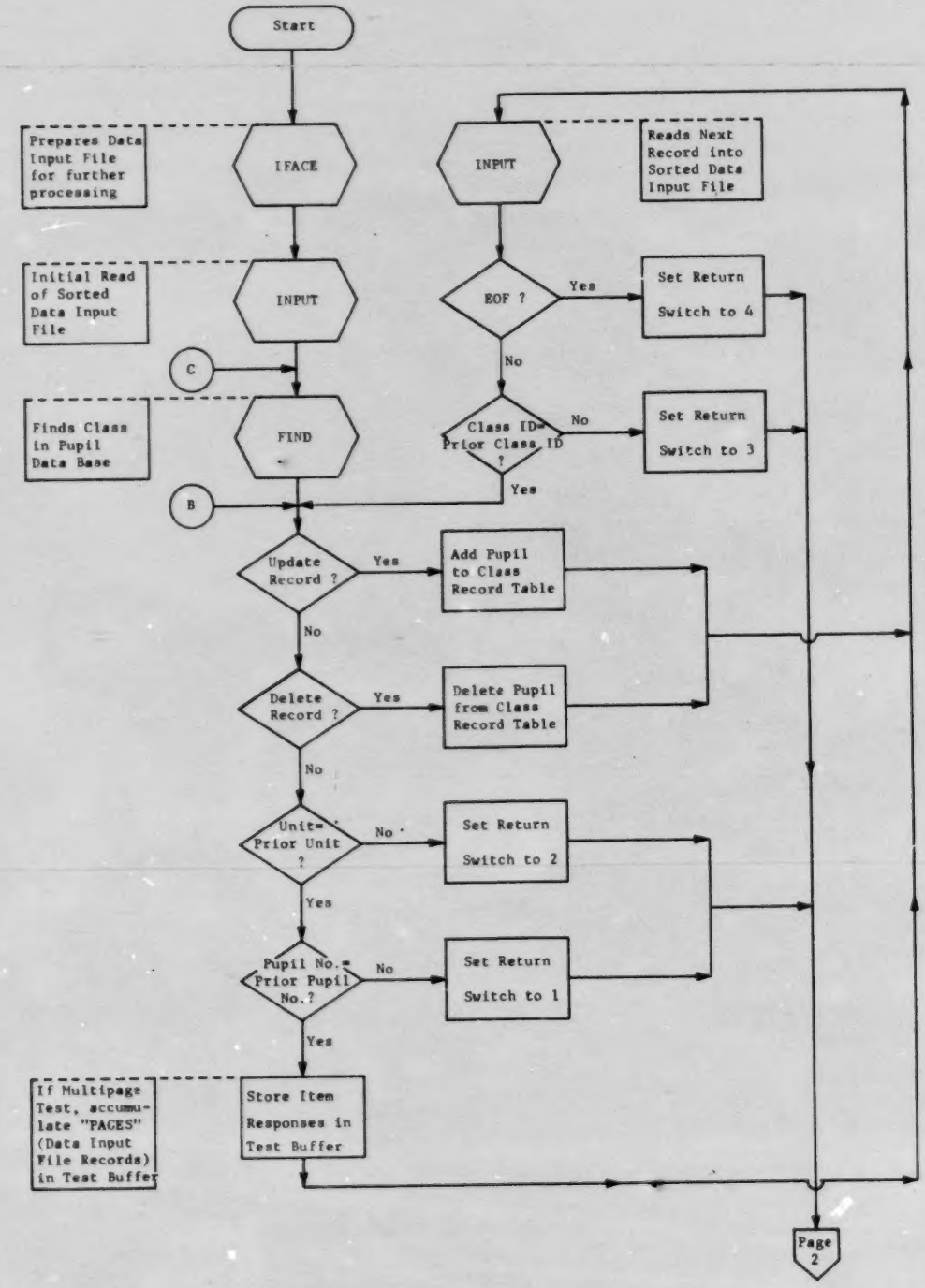
CIMS processes a file from UCLA's Remote Job Service. Exhibit 15 contains a listing of the control cards needed to run CIMS.

SWRI

FLOWCHART LAYOUT FORM

PROGRAMMER: Howard Wolfe
PROGRAM ID: C I M S

DATE: April 20, 1972
PAGE 1 OF 2

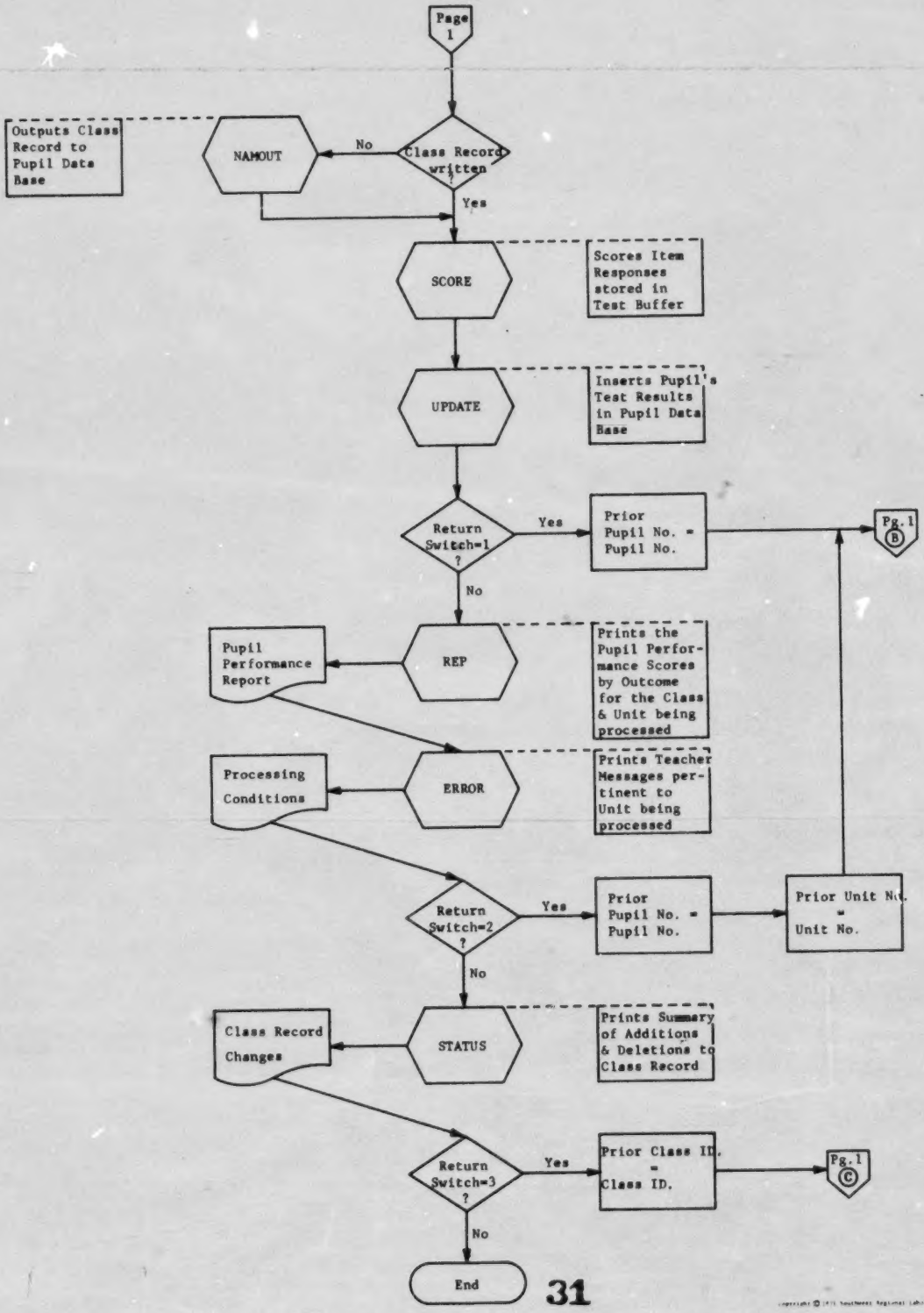


Copyright © 1972 Southwest Regional Laboratory

FLOWCHART LAYOUT FORM

PROGRAMMER: Howard Wolfe
PROGRAM ID: CIMS

DATE: April 20, 1972
PAGE 2 OF 2



Copyright © 1971 Southwest Regional Laboratory

9.0 - PROGRAM LISTING

```
      BLOCK DATA
      COMMON /IBLK2/  JD(6),ID(6)
      COMMON /IBLK3/ JDNAME(6,5),JSITE
      COMMON /IBLK11/ IERR(2,100),MREL
      COMMON /IBLK12/ ITYPE
      COMMON /IBLK22/ JHI
      COMMON /IBLK23/ ISWRL
      COMMON /IBLK24/ MSITE
      COMMON /JBLK3/  IRESP(28),IUNIT,IPROG,IPOS
      COMMON /JBLK25/LPROG,LUNIT,KEYBG,IOUTBG,ITMBG,KRIBG,
     $         ITOTBG,KRTBG,KEY(50),NOUT(50)
      COMMON /JBLK29/ ICODE,JCODE
      COMMON /KBLK4/  IREL,JREL,KREL,LREL
      COMMON /KBLK5/ ICLASS,IDELE,IADD,IPAGE,ITEST,JTEST,IREC,
     * JOLD,JNEW,JRIDE
      COMMON /LBLK1/ KD(6)
      DATA KD/6*0/
      DATA LUNIT/-99/,LPROG/-99/
      DATA MSITE/0/,JD/6*0/,ITYPE/-5/,ISWRL/6/,IRESP/28*' '/,
     *     IREL/0/,JREL/0/,KREL/0/,MREL/0/,LREL/0/,JHI/0/,
     *     ICLASS/0/,IDELE/0/,IADD/0/,IPAGE/0/,ITEST/0/,
     *     JTEST/0/ ,  JRIDE/0/,
     *     IREC/0/,JOLD/0/,JNEW/0/,ICODE/0/,JCODE/0/
      END
      SUBROUTINE IFACE
C  FOLLOWING LIST CONTAINS THE INPUT/OUTPUT FORMATS USED IN THE FOLLOWING
C READ, SORT, WRITE SUBROUTINE IFACE (690-CENTRAL COMPUTER INTERFACE)
C        FIELD NO    FIELD NAME                                  EDIT CODE  TOTAL
C              01    PORT NO.                                    I2              2
C              02    STUDENT NO                                  I2              4
C              03    PROGRAM NO                                  I2              6
C              04    UNIT NO                                     I2              8
C              05    PAGE NO                                     I1              9
C              06    DELETE INDICATOR                            I1             10
C              07    TEST INDICATOR                              I1             11
C              08    ITEM RESPONSES  01 THRU 04                  A4             15
C              09                    05 THRU 08                  A4             19
C              10                    09 THRU 12                  A4             23
C              11                    13 THRU 16                  A4             27
C              12                    17 THRU 20                  A4             31
C              13                    21 THRU 24                  A4             35
C              14                    25 THRU 28                  A4             39
C                   BLANK SPACE                                  1X             40
C              15    DISTRICT NO/SCHOOL NO/TEACHER NO            I6             46
C              16    UNUSED IN THIS VERSION OF IMS               I2             48
C              17    GRADE/CLASS                                 I4             52
C              18    DATE%YR/MO/DAY<                             I6             58
C              19    GROUP NO                                    I2             60
C              20    STUDENT NAME-FIRST FOUR LETTERS             A4             64
C              21                  SECOND FOUR LETTERS           A4             68
C              22    THIRD FOUR LETTERS                          A4             72
C              23    FOURTH FOUR LETTERS                         A4             76
C              24    FIFTH FOUR LETTERS                          A4             80
C
      INTEGER SORT
      EXTERNAL COMP
      COMMON /IBLK2/ JD(6),ID(6)
      COMMON /IBLK20/JYEAR,JMONTH,JDAY,JDENT,JTIME,JOVER/
```

```
      COMMON /KBLK1/ IADDR(500,6)
      DIMENSION KEY(500),KADDR(11,3)
      COMMON /KBLK9/ IBUFF(24,500)
      DIMENSION IHEAD(500)
      DIMENSION JBUFF(24)
      DIMENSION IWD1(5,2)
   40 FORMAT(' PORT NO. INVALID ON REC. NO.=',I5,' OF TRANS. FILE',///)
   30 FORMAT(' INPUT ERROR ON TRANS. FILE AT RECORD NO.=',I5///)
   50 FORMAT('1',24X,'IMS RUN LOG'////)
   51 FORMAT(' RUN DATE= ',I2,'/',I2,'/',I2//)
   52 FORMAT('      TIME= ',A4//)
   53 FORMAT('    RUN ID= ',A4//)
   54 FORMAT(' OVER-RIDE KEY IS ON'///)
   55 FORMAT(' END OF IFACE PROGRAM, COVER SHEET RECORDS= ',I6,' DATA RE
     *CORDS= ',I6///)
      DATA IYES/'Y'/
     *          '0   ','0*  ','00* ','000*','0000'/
      DATA KBLANK/'    '/
C  LEN IS THE LENGTH OF THE INPUT RECORD IN WORDS
C  FIRST READ DATE-IDENT-TIME-OVERRIDE CARD
C  K COUNTS STUDENT NAME, TEST AND DELETE RECORDS
      K=1
C  J COUNTS INPUT RECORDS
      J=0
      LEN=6
      CALL INPUT(4,2,JBUFF,LEN)
      IF(JBUFF(1).EQ.-3)GOTO 3000
      JMONTH=JBUFF(1)
      JDAY=JBUFF(2)
      JYEAR=JBUFF(3)
      JDENT=JBUFF(4)
      JTIME=JBUFF(5)
      JOVER=JBUFF(6)
      LEN=24
  100 J=J+1
      CALL INPUT(4,1,IBUFF(1,K),LEN)
C  IF EOF GOT 2000
      IF(IBUFF(1,K).EQ.-2)  GOTO              2000
C  IF READ ERROR GOTO 3000
      IF(IBUFF(1,K).EQ.-3)  GOTO              3000
      IPORT=IBUFF(1,K) + 1
C  VALID PORT NUMBERS ARE 1 THRU 10
      IF(IPORT.LT.1.OR.IPORT.GT.11) GOTO 4000
C  IF PROGRAM NUMBER IS ZERO  THIS IS A COVER FORM -GOTO 1000
      IF(IBUFF(20,K).NE.KBLANK) IBUFF(3,K)=32
       IF(IBUFF(3,K).EQ.0) GO TO 1000
C  IF THIS IS NOT A NAME FORM DATE IS OBTAINED FROM THE CLASS COVER FORM
C  IF PROGRAM NUMBER=32 THIS IS A NAME FORM. IN THIS CASE SET UNIT=-1
      IF(IBUFF(3,K).EQ.32) GOTO 127
C  OTHERWISE THIS IS A STUDENT NAME, TEST OR DELETE FORM
C  INSERT THE COVER SHEET ID, KEYS INTO THE RECORD
      IF(IBUFF(3,K).GT.5)GOTO398                                    04/16/
                                                                       HA
      IF(IBUFF(5,K).EQ.9)IBUFF(2,K)=0                               04/13/
      IF(IBUFF(3,K).GT.1.AND.IBUFF(3,K) .LT.5)GO TO 437
      GOTO 398
  437 IBUFF(5,K)=0
      IF(IBUFF(2,K).EQ.0) GOTO 398
      IBUFF(5,K)= 1 + (IBUFF(2,K) - 1) / 10
```

```
  398 IBUFF(15,K)=KADDR(IPORT,1)
C  SAVES DISTRICT/SCHOOL/TEACHER NUMBERS
      IBUFF(17,K)=KADDR(IPORT,2)
   SAVES GRADE/CLASS NUMBERS
      IBUFF(18,K)=KADDR(IPORT,3)
C  IF DELETE FIELD IS MARKED SET UNIT= -2/UNIT
C WHER '/' MEANS CHAIN.
C  PROVIDING NO ITEM RESPONSES ARE INDICATED
      IADDR(K,4)=IBUFF(4,K)                                          04/16/7
      IF(IBUFF(3,K).GT.5)GOTO123                                       CHG 1
      IF(IBUFF(7,K).EQ.1) GOTO                  123                  04/16/7
      IF(IBUFF(2,K).LT.1.OR.IBUFF(2,K).GT.60) IBUFF(6,K)=3           04/16/7
      IF(IBUFF(6,K).NE.1) GO TO 123
      ISW=1                                                          04/15/7
      IF(IBUFF(3,K).EQ.1.OR.IBUFF(3,K).EQ.5)ISW=2
      DO 500 MM=8,14
      DO 501 NN=1,4
      IF(IWD1(NN,ISW).EQ.IBUFF(MM,K)) GO TO 503
  501 CONTINUE
      IF(IWD1(5,ISW).EQ.IBUFF(MM,K)) GO TO 500
      IBUFF(6,K)=2
      GOTO 123
  500 CONTINUE
      IBUFF(6,K)=2
  502 GO TO 123
C  THIS CAUSES DELETES TO SORT LOW
  503 IADDR(K,4)=-2*100-IBUFF(4,K)
C  SAVE THE SORT KEYS FOR THIS RECORD
  123 IADDR(K,1)=KADDR(IPORT,1)
C  THIS SAVES THE DISTRICT/SCHOOL/TEACHER NUMBERS FOUND ON THE COVER REC
      IADDR(K,2)=KADDR(IPORT,2)
C  THIS SAVES THE GRADE/CLASS NUMBERS FOUND ON THE COVER FORM
      IADDR(K,3)=0
C  NORMALLY THIS WOULD BE PROGRAM NUMBER BUT IN THIS VERSION OF IMS PROG
C RAM NUMBER WILL NOT BE SORTED ON
      IADDR(K,5)=IBUFF(2,K)
C  THIS SAVES THE STUDENT NUMBER
C  THIS SAVES THE PAGE NUMBER
       K=K+1
      IF(K.EQ.500) CALL ERR(&2000,-6,0,0,0,0)
      GOTO 100
  127 IADDR(K,1)=IBUFF(15,K)
      IADDR(K,2)=IBUFF(17,K)
      IADDR(K,3)=0
      IADDR(K,4)=-1
      IADDR(K,5)=IBUFF(2,K)
      IADDR(K,6)=IBUFF(5,K)
      K=K+1
      IF(K.EQ.500) CALL ERR(&2000,-6,0,0,0,0)
      GOTO  100
 1000 KADDR(IPORT,1)=IBUFF(15,K)
C  THIS SAVES THE DISTRICT/SCHOOL/TEACHER NUMBERS
      KADDR(IPORT,2)=IBUFF(17,K)
   THIS SAVES THE GRADE/CLASS NUMBERS
      KADDR(IPORT,3)=IBUFF(18,K)
C  THIS SAVES THE DATE
      GOTO 100
 2000  K=K-1
```

```
      J=J-1
      I=SORT(KEY,IHEAD,K,COMP)
 2001 IF(I.EQ.0) GOTO 5000
      LEN=24
      CALL OUTPUT(4,1,IBUFF(1,I),LEN)
      I=KEY(I)
      GOTO 2001
 3000 WRITE(6,30) J
C  FIND NEXT COVER SHEET
  777 IF(J.EQ.0) STOP
      J=J+1
      CALL INPUT(4,1,IBUFF(1,K),LEN)
      IF(IBUFF(1,K).EQ.-2) GOTO 2000
      IF(IBUFF(1,K).EQ.-3) GOTO 3000
      IPORT=IBUFF(1,K) + 1
      IF(IPORT.LT.1.OR.IPORT.GT.11) GOTO 4000
      IF(IBUFF(20,K).NE.KBLANK) IBUFF(3,K)=32
      IF(IBUFF(3,K).EQ.0) GOTO 1000
      GOTO 777
C  NOTE THAT SEVERAL CLASSES MAY LOSE DATA RECORDS DURING THIS
C  RECOVERY ROUTINE
 4000 WRITE(6,40) J
      GOTO 777
 5000 WRITE(6,50)
      WRITE(6,51)  JMONTH,JDAY,JYEAR
      WRITE(6,52) JTIME
      WRITE(6,53) JDENT
      IF(JOVER.EQ.IYES) WRITE(6,54)
      KK=J-K
      WRITE(6,55) KK,K
      END FILE 3
      REWIND 3
      RETURN
      END
      SUBROUTINE IOIBM
      COMMON /IBLK12/ ITYPE
 1316 FORMAT(I1,1X,3(4A4,A3),I2,2X,I2,1X,6I2,2X)
 9316 FORMAT(1X,I1,1X,3(4A4,A3),I2,2X,I2,1X,6I2,2X)                    CHG 1
 1318 FORMAT(I1,1X,5A4,1X,A1,35X,I2,2X,A1,I2,12X,I2)
 1319 FORMAT(I1,1X,6I2,1X,I3,1X,28A1,1X,3I2,5X,I2,1X,2I2,12X,I2)
 1324 FORMAT(I1,19A4,A3)
 1346 FORMAT(4I2,3I1,7A4,1X,I6,I2,I4,I6,I2,5A4)
 1347 FORMAT(8X,3I2,4X,A4,6X,A4,10X,A1)
 1361 FORMAT(4I2,3I1,28A1,1X,6I2,4I2,5A4)
      ENTRY INPUT(IFILE,IREC,IBUFF,LEN)
      DIMENSION IBUFF(LEN)
      GO TO (1,2,3,4,5,6),IFILE
    1 CONTINUE
    2 GO TO (1016,1017,1018,1019,1020,1021,1022,1023,1024),IREC
    3 GOTO(1031,1032,1033),IREC
    4 GO TO(1046,1047), IREC
    5 GO TO (1061),IREC
    6 CONTINUE
 1016 READ(4,1316) IBUFF
      ITYPE=IBUFF(1)
      GOTO 99
 1017 GOTO 1016
 1018 READ(4,1318) IBUFF
```

```
      ITYPE=IBUFF(1)
      GO TO 99
 1019 READ(4,1319) IBUFF
      ITYPE=IBUFF(1)
      GOTO 99
 1020  CONTINUE
 1021 CONTINUE
 1022 CONTINUE
 1023 CONTINUE
 1024 READ(4,1324) IBUFF
      ITYPE=IBUFF(1)
      GO TO 99
 1031 IUNIT=IBUFF(1)
      IPOS=IBUFF(2)
      CALL NTRAN(IUNIT,6,1,IBUFF,IPOS)
      GOTO 99
 1032 IUNIT=IBUFF(1)
      CALL NTRAN(IUNIT,2,LEN,IBUFF,0)
      GOTO 99
 1033 IUNIT=IBUFF(1)
      CALL NTRAN(IUNIT,10,1,IBUFF,0)
      GO TO 99
 1046 READ(5,1346,END=301,ERR=401) IBUFF
      IBUFF(5)=IABS(IBUFF(5))
      IBUFF(6)=IABS(IBUFF(6))
      IBUFF(7)=IABS(IBUFF(7))
      GO TO 99
 1047 READ(5,1347,ERR=401) IBUFF
      GOTO 99
 1061 READ(3,1361,END=301,ERR=401) IBUFF
      GOTO 99
      ENTRY OUTPUT(IFILE,IREC,IBUFF,LEN)
      GOTO(501,502,503,504),IFILE
  501 GOTO(2501),IREC
  502 GO TO (2516,2517,2518,2519,2520,2521,2522,2523,2524),IREC
  503 CONTINUE
  504 GO TO (2546),IREC
 2501 CONTINUE
 2516 WRITE(6,9316) IBUFF                                          CHG 1
      WRITE(8,1316) IBUFF
      ITYPE=-IBUFF(1)
      GO TO 99
 2517 GOTO 2516
 2518 WRITE(8,1318) IBUFF                                          CHG 1
      ITYPE=-IBUFF(1)
      GO TO 99
 2519 WRITE(8,1319)  IBUFF
      ITYPE=-IBUFF(1)
      GOTO 99
 2520 GO TO 2519
 2521 CONTINUE
 2522 CONTINUE
 2523 CONTINUE
 2524 WRITE(8,1324) IBUFF
      ITYPE=-IBUFF(1)
      GOTO 99
 2546 WRITE(3,1346) IBUFF
   99 RETURN
```

```
  301 IBUFF(1)=-2
      GO TO 99
  401 IBUFF(1)=-3
      GO TO 99
      END
      INTEGER FUNCTION SORT(LISTS,IHEAD,N,COMPAR)
C
C     GENERAL SORT ROUTINE
C     TO BE SORTED.
C     LISTS IS AT LEAST N WORDS LONG, WHERE N IS THE NUMBER OF RECORDS
C     IHEAD IS ALSO N WORDS LONG AND IS USED AS A SCRATCH AREA.
C     COMPAR IS A 2-ARGUMENT INTEGER FUNCTION HAVING A NEGATIVE VALUE
C     IF THE RECORD REFERENCED BY THE FIRST ARGUMENT IS TO BE SORTED
C     BEFORE THE RECORD REFERENCED BY THE SECOND ARGUMENT, A ZERO VALUE
C     IF THE ORDER IS IMMATERIAL, AND A POSITIVE VALUE OTHERWISE.
C     SINCE SORT DEALS WITH RECORD INDEXES ONLY, THE SORT KEYS MAY BE
C     ARBITRARILY COMPLEX, AND THE RECORDS TO BE SORTED NEED NOT BE
C     IN MEMORY.
C
C     USAGE
C     INTEGER SORT
C     EXTERNAL COMPAR (OPTIONAL, COMPAR MAY BE INTERNAL)
C     I=SORT(LIST,IHEAD,N,COMPAR)
C   A IF (I.EQ.0) GO TO B
C     DO ANY PROCESSING FOR RECORD I HERE.
C   B CONTINUE
C
      INTEGER COMPAR,DIRECT
      INTEGER LISTS(N),IHEAD(N)
      EQUIVALENCE (MP,DIRECT),(JHEAD,NCHAIN)
C     I=LIST(I)
C     GO TO A
      EXTERNAL COMPAR
C
C     START BY SCANNING ALL RECORDS AND DETECTING ASCENDING OR
C     DESCENDING CHAINS.  PUT THE HEADS OF THESE CHA2NS IN HEAD.
C
      M=1
C     INDECATE BEGINNING OF CHAIN
      DIRECT=0
      I=1
C     INDICATE RECORD 1 HEAD OF CHAIN
      JHEAD=1
      LISTS(1)=0
1     IF(I.GE.N) GO TO 10
      I1=I+1
C
C     CHECK SEQUENCE OF TWO ADJACENT RECORDS.
C
      IF(COMPAR(I,I1)) 2,22,8
22    IF(DIRECT) 9,3,4
C
C     RECORDS I,I+1 ARE IN ASCENDING SEQUENCE.
C
2     IF(DIRECT) 7,3,4
C
C     START A FORWARD CHAIN.
C
```

```
3       DIRECT=1
        JHEAD=I
C
C       CONTINUE A FORWARD CHAIN.
C
4       LISTS(I)=I1
5       I=I1
        GO TO 1
C
C       HAD A CHAIN, GOT A SEQUENCE CHANGE.
C
C       INDICATE END OF CHAIN
6       LISTS(I)=0
C       SAVE HEAD OF CHAIN
7       IHEAD(M)=JHEAD
        M=M+1
        DIRECT=0
        GO TO 5
C
C       RECORDS I,I+1 ARE IN DESCENDING SEQUENCE.
C
8       IF(DIRECT) 9,99,6
C
C       START A BACKWARD CHAIN.
C
99      DIRECT=-1
C       INDECATE END OF CHAIN
        LISTS(I)=0
C
C       CONTINUE A BACKWARD CHAIN
C
9       LISTS(I1)=I
C       UPDATE HEAD OF CHAIN
        JHEAD=I1
        GO TO 5
C
C       END OF CHAIN DETECTION,  CLEAN UP AND START MERGE PASS.
C
10      IHEAD(M)=JHEAD
        IF (DIRECT .GE. 0) LISTS(I)=0
        IF (DIRECT .EQ. 0) IHEAD(M)=N
11      I=-1
        MP=0
12      I=I+2
        IF(I.LT.M) GO TO 15
        IF(I.EQ.M) GO TO 14
13      M=MP
        IF(M.NE.1) GO TO 11
        SORT=IHEAD(1)
        IHEAD(1)=0
        RETURN
14      MP=MP+1
        IHEAD(MP)=IHEAD(I)
        IF(I.EQ.1) GO TO 13
        IHEAD(I)=0
        GO TO 13
15      J1=IHEAD(I)
        J2=IHEAD(I+1)
```

```
      IHEAD(I)=0
      IHEAD(I+1)=0
      I1=J1
      I2=J2
      MP=MP+1
      NCHAIN=0
16    IF(COMPAR(J1,J2).LE.0) GO TO 17
C
C     J2 SHOULD COME BEFORE J1.
C
      IF (NCHAIN .EQ. 0) IHEAD(MP)=J2
      IF (NCHAIN .EQ. 1) LISTS(I1)=J2
      NCHAIN=2
      I2=J2
      J2=LISTS(I2)
      IF(J2.NE.0) GO TO 16
      LISTS(I2)=J1
      GO TO 12
C
C     J1 SHOULD COME BEFORE J2.
C
17    IF (NCHAIN .EQ. 0) IHEAD(MP)=J1
      IF (NCHAIN .EQ. 2) LISTS(I2)=J1
      NCHAIN=1
      I1=J1
      J1=LISTS(I1)
      IF(J1.NE.0) GO TO 16
      LISTS(I1)=J2
      GO TO 12
      END
      INTEGER FUNCTION COMP(I,J)
      COMMON /KBLK1/ IADDR(500,6)
      COMMON /KBLK9/ IBUFF(24,500)
      DO 100 N=1,6
      IF(N.EQ.3) GO TO 100
      IF(IADDR(I,N)-IADDR(J,N)) 10,100,30
   10 COMP=-1
      RETURN
   30 COMP=1
      RETURN
  100 CONTINUE
      COMP=0
      IF(IBUFF(3,I).GT.1.AND.IBUFF(3,I).LT.5) GOTO 200
      IF(IBUFF(3,I).EQ.1.OR.IBUFF(3,I).EQ.5) GOTO 199
      RETURN
  199 IBUFF(1,I)=-8
      IBUFF(1,J)=-8
      RETURN
  200 IBUFF(1,I)=-9
      IBUFF(1,J)=-9
      RETURN
      END
        SUBROUTINE SPACE
      COMMON /IBLK23/ ISWRL
      COMMON /JBLK7/ IPAGE,LINES
90 J  FORMAT('==',1X,I2)
9012  FORMAT(66(' '))
      IDIFF=62-LINES
```

04/16/
04/16/

```
      WRITE(ISWRL,9010) IDIFF
      WRITE(ISWRL,9012)
      LINES=0
      RETURN
      END
 1318 FORMAT(I1,1X,5A4,1X,A1,35X,I2,2X,A1,I2,12X,I2)                    CHG 15
 1460 FORMAT(1X,66('-')/' EOF-ID= ',6I2,' JD= ',6I2)
 1462 FORMAT(' OLD SDB COUNT= ',I6/
     *'          NAMES=',I6/'          TESTS=',I6/
     *' NEW SDB COUNT= ',I6///)
 1720 FORMAT(' UNIT= ',I2,' TESTS= ',I2,' DELETIONS= ',I2,
     *' ADDITIONS= ',I2///)
 1725 FORMAT(1X,66('-')/' BEGIN CLASS ID= ',6I2///)
      COMMON /IBLK2/ JD(6),ID(6)
      COMMON /IBLK11/ IERR(2,100),MREL
      COMMON/IBLK12/ ITYPE
      COMMON /IBLK20/JYEAR,JMONTH,JDAY,JDENT,JTIME,JOVER
      COMMON /IBLK21/ IYEAR,IMONTH,IDAY
      COMMON /IBLK22/ JHI
      COMMON /IBLK23/ ISWRL
      COMMON /IBLK24/ MSITE
      COMMON /JBLK3/  IRESP(28),IUNIT,IPROG,IPOS
      COMMON /JBLK4/ ISCORE(60,10)
      COMMON /JBLK30/ IEBUF(11,60)
      COMMON /KBLK2/  ISIG(60),JSIG(60),KSIG(60),LSIG(60)
      COMMON /KBLK4/  IREL,JREL,KREL,LREL
      COMMON /KBLK5/  ICLASS,IDELE,IADD,IPAGE,ITEST,JTEST,
     *                IREC,JOLD,JNEW,JRIDE
      COMMON K4
      DEFINE FILE 14(1000,336,L,K4)
      DIMENSION IBUFF(50)
      DATA IBLANK/' '/ ,ISTAR/'*'/
      DATA IZERO/'0'/
C  THE FOLLOWING LIST CONTAINS THE READ-IN FORMAT USED IN THE FOLLOWING
C SUBROUTINE
C       FIELD NO   FIELD NAME                                  EDIT CODE  TOTAL COL
C       01   FLAG FROM IFACE SUBPROGRAM
C            01    -8 INDICATES A MULTIPLE FYCSP SHEET          I2            2
C                  -9 INDICATES A MULTIPLE LMS SHEET
C                  OTHERWISE THIS IS PORT NUMBER
C            02    STUDENT NO                                   I2            4
C            03    PROGRAM NO                                   I2            6
C            04    UNIT NO                                      I2            8
C            05    PAGE NO                                      I1            9
C            06    DELETE INDICATOR                             I1           10
C            07    TESTED INDICATOR                             I1           11
C            08    ITEM RESPONSE 1                              A1           12
C            09    ITEM RESPONSE 2                              A1           13
C            ....
C            35    ITEM RESPONSE 28                             A1           39
C                  BLANK SPACE                                  1X           40
      CONTINUE
C            36    DISTRICT NO                                  I2           42
C            37    SCHOOL NO                                    I2           44
C            38    TEACHER NO                                   I2           46
C            39    PROGRAM NO                                   I2           48
C            40    GRADE NO                                     I2           50
C            41    CLASS NO                                     I2           52
```

```
C              42   YEAR                                              I2          54
C              43   MONTH                                             I2          56
C              44   DAY                                               I2          58
C              45   GROUP NO                                          I2          60
C              46   STUDENT NAME LETTERS  1 - 4                       A4          64
C              47   STUDENT NAME LETTERS  5 - 8                       A4          68
C              48   STUDENT NAME LETTERS  9 -12                       A4          72
C              49   STUDENT NAME LETTERS 13 -16                       A4          76
C              50   STUDENT NAME LETTERS 17 -20                       A4          80
C  CORE CAN BE SAVED BY RUNNING IFACE AS STEP 1 OF A 2 STEP JOB
      CALL                                          IFACE
C  INITIALIZE FOR RUN
      IPTR=0
      ISWRL=7
      IOK=-2
C  INITIALIZE INPUT BUFFER
C  INPUT BUFFER LENGTH IN WORDS
  100 LEN=50
      CALL INPUT(5,1,IBUFF,LEN)
      IF(IBUFF(1).EQ.-2)  GOTO 460                                      CHG 07
      IREC=IREC+1
C  BEGIN CLASS
C  SET IDENTIFICATION KEYS- DISTRICT/SCHOOL/TEACHER/PROGRAM/GRADE/CLASS
  105 DO 110 I=1,6
  110 ID(I)=IBUFF(I+35)
      ISW1=0
C  FIND CLASS IN STUDENT DATA BASE
      CALL                                          FIND(IER)
C  IF IER=1 CLASS WAS NOT FOUND IN STUDENT DATA BASE
      IF(IER.EQ.1) GOTO                             100
      WRITE(6,1725) JD                                                  04/15/7
      IPROG=JD(4)
      CALL                                          NAMIN
C  LNAM=0 INDICATES THAT STUDENT NAME BLOCK IN THE STUDENT DATA BASE HAS
C NOT YET BEEN WRITTEN TO THE NEW SDB, OTHERWISE LNAM=1.
      LNAM=0
C  BEGIN UNIT
  115 IUNIT=IBUFF(4)
      DO 1601 KK=1,60
      DO 1601 MM=1,10
 1601 ISCORE(KK,MM) =0
  116 INO=IBUFF(2)                                                      CHG 1
      IYEAR=IBUFF(42)                                                   CHG 1
      IMONTH=IBUFF(43)                                                  CHG 1
      IDAY=IBUFF(44)
      GOTO                                          630
C  INPUT RECORD
  600 LEN=50
      CALL INPUT(5,1,IBUFF,LEN)
C  IF THIS IS EOF GO TO                           450
      IF(IBUFF(1).EQ.-2) GOTO                       450
      IREC=IREC+1
C  IS CLASS ID FOR THIS RECORD THE SAME AS THE INITIAL RECORD FOR THIS CLASS
      DO 620 I=1,6
      IF(IBUFF(I+35).NE.ID(I)) GOTO                 300
  620 CONTINUE
  630 IF(IBUFF(6).EQ.1.AND.IBUFF(7).NE.1)GOTO 147                       04/15/7
      IF(IBUFF(3).EQ.32) GOTO 250                                       04/16/7
```

```
      IF(ISW1.EQ.0) GOTO                          349                    04/15/7
      IF(IBUFF(4).NE.IUNIT) GOTO                  350                    04/15/7
      IF(IBUFF(2).NE.INO) GOTO                    400                    04/15/7
  147 IF(IYEAR.EQ.0.OR.IMONTH.EQ.0.OR.IDAY.EQ.0)                         04/15/7
     *CALL ERR(&631,-8,IBUFF(2),0,0,0)
  631 IF(IBUFF(3).LT.1.OR.IBUFF(3).GT.5) CALL ERR(&600,26,0,0,0,0)       04/16/7
C  IF THIS IS AN LMS SHEET AND PAGE NUMBER NOT CODED                       CHG 0
      IF(IBUFF(3).GT.1.AND.IBUFF(5).EQ.0.AND.IBUFF(3).LT.5)              04/13/7
     *CALL ERR(&600, 6,IBUFF(5),0,0,0)                                    CHG 0
C  IF THIS IS A 'MULTIPLE' LMS SHEET                                       CHG 0
      IF(IBUFF(1).EQ.-9.AND.IBUFF(5).NE.0)                                CHG 0
     *CALL ERR(&600, 9,IBUFF(5),0,0,0)                                    CHG 0
      IF(IBUFF(7).EQ.1.AND.IBUFF(6).EQ.1)                                 CHG 1
     *CALL ERR(&150,19,IBUFF(2),0,0,0)                                    CHG 1
      IF(IBUFF(6).EQ.1) GOTO                      200                    CHG 0
C  TEST RECORD
  150 IF(IBUFF(2).LT.1)                                                  04/15/7
     *CALL ERR(&600,21,IBUFF(2),0,0,0)
      IF(IBUFF(2).GT.60)
     *CALL ERR(&600,25,IBUFF(5),0,0,0)
C  IF THIS IS A 'MULTIPLE' FYCSP SHEET PRINT ERROR MESSAGE
      IF(IBUFF(1).EQ.-8.AND.IBUFF(5).EQ.1)
     *CALL ERR(&600,22,IBUFF(2),0,0,0)
      IF(IBUFF(1).EQ.-8)
     *CALL ERR(&600,23,IBUFF(2),0,0,0)
C  IF THIS IS TEACHER MARKED FORM AND IT CONTAINS MARKS BUT THE TESTED COLUMN
C IS NOT MARKED THEN AN ERROR HAS OCCURRED
      IF(IBUFF(7).NE.1.AND.IBUFF(3).GT.1.AND.IBUFF(3).LT.5)              04/13/7
     *CALL ERR(&159,16,IBUFF(2),0,0,0)
  159 IF(IBUFF(2).GT.JHI.AND.IBUFF(3).GT.1.AND.IBUFF(3).LT.5)            04/13/7
     *GO TO 158
      IF(IBUFF(2).GT.JHI)
     *CALL ERR(&600,14,IBUFF(2),0,0,0)
      IF(IEBUF(9,INO).NE.IBLANK.AND.IBUFF(3).GT.1.AND.IBUFF(3).LT.5)     04/13/7
     *CALL ERR(&600,12,IBUFF(2),0,0,0)
      IF(IEBUF(9,INO).NE.IBLANK)
     *CALL ERR(&600,15,IBUFF(2),0,0,0)
      GO TO 156
  158 MHI=10*(1+(JHI-1)/10)
      I=1 + (IBUFF(2)-1)/10
      IF(IBUFF(2).GT.MHI) CALL ERR(&600,13,I,0,0,0)
      CALL ERR(&600,11,IBUFF(2),0,0,0)
C  IF DELETE FIELD MARKED TOGETHER WITH SCORES
  156 IF(IBUFF(6).EQ.2)
     *CALL ERR(&155,17,IBUFF(2),0,0,0)
C  MOVE THE ITEM RESPONSES ON A TEST PAGE TO BUFFER IRESP WHERE COMPLETE
C TEST IS STORED
  155 DO 160 I=1,28
C  AN ASTERISK IS STORED IMMEDIATELY BEHIND THE LAST ITEM RESPONSE ON
C PAGE BY THE 690 SOFTWARE
      IF(IBUFF(7+I).EQ.ISTAR) GOTO                163
C  IF THERE ARE MORE THAN 28 RESPONSES/TEST A SYSTEM ERROR HAS OCCURRED
C HOWEVER THIS PARAMETER (28) CAN BE INCREASED BY MODIFYING THE SOURCE
C PROGRAM
      IF(IPTR.EQ.28)
     *CALL ERR(&500,-3,IBUFF(2),0,0,0)
      IPTR=IPTR+1
  160 IRESP(IPTR)=IBUFF(I+7)
```

```
C  IOK=-1 INDICATES THAT STUDENT TEST BUFFER CONTAINS SOME UNSCORED
C  ITEMS.
  163 IOK=-1
      GOTO                          600
C  DELETE STUDENT
C  IF DELETED STUDENT NUMBER IS NOT IN CLASS A TEACHER ERROR HAS OCCURRED
  200 IF(IBUFF(2).GT.60)
     *CALL ERR(&600,2,IBUFF(2),0,0,0)                                    U
      IF(IBUFF(2).GT.JHI)
     *CALL ERR(&600,4,IBUFF(2),0,0,0)                                    U
      IF(IBUFF(2).LT.1)
     *CALL ERR(&600,1,IBUFF(2),0,0,0)                                    U
      IF(JREL.EQ.0) GOTO                          210
C  IF THE SAME STUDENT HAS ALREADY BEEN DELETED IN THIS RUN A TEACHER
C ERROR HAS OCCURRED
      IF(JSIG(JREL).EQ.IBUFF(2))
     *CALL ERR(&600,3,IBUFF(2),0,0,0)                                    U
  210 KNO=IBUFF(2)
C  IF THE STUDENT IS ALREADY INACTIVE A TEACHER ERROR HAS OCCURRED
      IF(IEBUF(9,KNO).NE.IBLANK)                                         A
     *CALL ERR(&600,4,IBUFF(2),0,0,0)                                    U
C  INACTIVE CODE
      IEBUF(9,KNO)=IZERO
      JREL=JREL+1
      JSIG(JREL)=KNO
      IDELE=IDELE+1
      GOTO                          600
C  NEW STUDENT
C  JHI IS SET IN NAME SUBROUTINE ENTRY POINT NAMIN AND INDICATES THE
C HIGHEST ASSIGNED STUDENT NUMBER IN THE CLASS
C  INCREMENTING BY 1 ASSIGNS A NEW STUDENT NO
  250 JHI=JHI+1
C  IF JHI > 60 A SYSTEM ERROR HAS OCCURRED
      DO 251 L=46,50                                                     U
      IF(IBUFF(L).NE.IBLANK) GO TO 252                                   U
  251 CONTINUE                                                           U
      CALL ERR(&252,-7,JHI,0,0,0)
  252 IF(JHI.GT.60)                                                      U
     *CALL ERR(&600,-5,0,0,0,0)
      KREL=KREL+1
C  KSIG IS A LIST OF NEW STUDENT NUMBERS
      KSIG(KREL)=JHI
C  STUDENT NAME
      DO 260 I=1,5
  260 IEBUF(I+1,JHI)=IBUFF(45+I)
C  RECORD TYPE
      IEBUF(1,JHI)=3
C  LAST NAME CARD INDICATOR
      IEBUF(7,JHI)=IBLANK
C  COMMAND CODE (NOT USED IN THIS PROGRAM)
      IEBUF(8,JHI)=0
C  ACTIVE CODE
      IEBUF(9,JHI)=IBLANK
C  GROUP NO
      IEBUF(10,JHI)=IBUFF(45)
C  N J STUDENT NUMBER
      IEBUF(11,JHI)=JHI
      IADD=IADD+1
```

```
      GOTO                                600
C  IRET=3 ESTABLISHES RETURN FROM STUDENT TEST PROCESSING WHICH BEGINS
C AT STATEMENT 545
  300 IRET=3
      GOTO                          700
  349 ISW1=1
      IUNIT=9999
      IRET=5
      GOTO 700
  350 IRET=2
      GOTO                                  700
  400 IRET=1
      GOTO                                700
  450 IRET=4
      GOTO                      700
  460 DO 1667 LL=1,6
 1667 ID(LL)=99
      CALL FIND(IER)
      WRITE(6,1460) ID,JD
      WRITE(6,1462) JOLD,IADD,ITEST,JNEW
      STOP
  700 IF(LNAM.NE.0) GOTO                          710
C  LNAM=0 INDICATES THAT STUDENT NAME BLOCK IN THE STUDENT DATA BASE HAS
C NOT YET BEEN WRITTEN TO THE NEW SDB, OTHERWISE LNAM=1,
      CALL                                        NAMOUT
      DO 9991 LMN=1,JHI                                                  CHG 15
 9991 WRITE(6,1318) (IEBUF(JJ,LMN),JJ=1,11)                              CHG 15
      LNAM=1
C  IOK=-1 INDICATES IT IS OK TO SCORE STUDENT ALTHOUGH
C SOME SCORES MAY BE MISSING
  710 IF(IOK.NE.-1) GOTO                          715
C  IREL IS AN INCREMENTAL COUNTER OF STUDENT TEST RECORDS
      IREL=IREL+1
C  ISIG IS A LIST OF STUDENTS FOR WHICH TESTS HAVE BEEN SUBMITTED
      ISIG(IREL)=INO
C  IPOS IS USED BY SCORE SUBROUTINE
      IPOS=IREL
      CALL                                        SCORE
      IF(IPOS.EQ.IREL) CALL UPDATE
      IF(IPOS.EQ.IREL) ITEST=ITEST+1
  715 IF(IPTR.EQ.0) IPTR=28
C  BLANK TEST RESPONSES BUFFER
      DO 720 I=1,IPTR
  720 IRESP(I)=IBLANK
      IPTR=0
      IOK=-2
C  RETURN TO APPROPRIATE POINT
      IF(IRET.EQ.1)GOTO                            116
      WRITE(6,1720) IUNIT,IREL,JREL,KREL
      IF(IRET.EQ.5) GOTO 115
C  IF IREL IS NOT EQUAL 0 THERE ARE TEST SCORES TO BE REPORTED
      IF(IREL.NE.0) CALL                          REP
      IREL=0
C  IF MREL IS NOT EQUAL 0 THERE ARE TEACHER ERRORS TO BE REPORTED
      IF(MREL.NE.0) CALL ERROR
      LREL=0
      MREL=0
      IF(IRET.EQ.2) GOTO                          115
```

```
C  IF JREL OR KREL ARE NOT EQUAL 0 THERE ARE NEW AND/OR DELETED STUDENTS
C TO BE REPORTED
      IF(JREL.NE.0.OR.KREL.NE.0) CALL          STATUS
      JREL=0
      KREL=0
      IF(IRET.EQ.3) GOTO                       105
      GOTO                                     460
      END
      INTEGER FUNCTION COMP1(I,J)
      COMMON /KBLK2/ ISIG(60),JSIG(60),KSIG(60),LSIG(60)
      IF(ISIG(I)-ISIG(J)) 10,20,30
10    COMP1=-1
      RETURN
20    COMP1=0
      RETURN
30    COMP1=1
      RETURN
      END
      INTEGER FUNCTION COMP4(I,J)
      COMMON /IBLK11/ IERR(2,100),MREL
      DO 110 N=1,2
      IF(IERR(N,I)-IERR(N,J)) 100,110,120
  110 CONTINUE
      COMP4=0
      RETURN
  100 COMP4=-1
      RETURN
  120 COMP4=1
      RETURN
      END
      SUBROUTINE SCORE
      IMPLICIT INTEGER (A-Z)
C  CHECK FOR ERRORS, THEN SCORE THE ITEM RESPONSE ARRAY FOR ONE STUDENT
      COMMON /IBLK2/ JD(6),ID(6)
      COMMON/IBLK7/NIOUT(10),LEVCRI(10),KRITOT,ITOT,NOCTOT,IPACK(10)
      COMMON /IBLK11/ IERR(2,100),MREL
      COMMON /JBLK3/ IRESP(28),NUNIT,NPROG,NPOS
      COMMON /JBLK4/ SCORES(60,10)
      COMMON /JBLK25/LPROG,LUNIT,KEYBG,IOUTBG,ITMBG,KRIBG,
     $         ITOTBG,KRTBG,KEY(50),NOUT(50)
      COMMON /KBLK2/ ISIG(60),JSIG(60),KSIG(60),LSIG(60)
      COMMON /KBLK4/ IREL,JREL,KREL,LREL
      DIMENSION IWORD(84),LPACK(10)
      DATA IBLANK/' '/
C  BYPASS THE LOADING OF KEY IF ITS ALREADY LOADED
      IF(NPROG.EQ.LPROG.AND.NUNIT.EQ.LUNIT)GO TO 2
            LPROG=NPROG
            LUNIT=NUNIT
C  REWIND DRUM TO REFERENCE POINT 0
      IWORD(1)=14
      LEN=1
      CALL INPUT(3,3,IWORD,LEN)
C  POSITION DRUM TO PROPER PROGRAM AND UNIT LOCATION
      LEN=2
      IWORD(2)=((NPROG-1)*20*3+(NUNIT-1)*3)
      IF(IWORD(2).LT.0) CALL ERR(826,26,0,0,0,0)
      IWORD(1)=14
      CALL INPUT(3,1,IWORD,LEN)
```

```
C  READ ANSWER KEY FROM INTERNAL FILE
      LEN=84
      IWORD(1)=14
      CALL INPUT(3,2,IWORD,LEN)
C  STORE THE NEWLY LOADED IWORD ARRAY INTO THE TEST DESCRIPTOR VARIABLE NAMES
   83 LPROG=IWORD(1)
      LUNIT=IWORD(2)
C  DETERMINE WHETHER OR NOT THE CORRECT KEY WAS LOADED
      IF (NPROG.EQ.LPROG.AND.NUNIT.EQ.LUNIT) GO TO 88
      CALL ERR(826,26,0,0,0,0)
   88 KEYBG=IWORD(3)
      IOUTBG=IWORD(4)
      ITMBG=IWORD(5)
      KRIBG=IWORD(6)
      KRTBG=IWORD(8)
      NOCTOT=KRIBG-ITMBG
      ITOT=IWORD(ITOTBG)
      KRITOT=IWORD(KRTBG)
      DO 700 I=1,6
      IPACK(I)=0
  700 LPACK(I)=0
      DO 100 NITEM=1,ITOT
        KEY(NITEM)=IWORD(KEYBG+NITEM-1)
        NOUT(NITEM)=IWORD(IOUTBG+NITEM-1)
      IX=NOUT(NITEM)
      LPACK(IX)=1
  100 CONTINUE
      J=1
      DO 702 I=1,6
      IF(LPACK(I).NE.0) GOTO 704
      GOTO 702
  704 IPACK(J)=I
      J=J+1
  702 CONTINUE
      J=1
      DO 703 I=1,6
      IF(LPACK(I).NE.0)LPACK(I)=J
      J=J+1
  703 CONTINUE
      DO 200 NOC=1,NOCTOT
       NIOUT(NOC)=IWORD(ITMBG+NOC-1)
       LEVCRI(NOC)=IWORD(KRIBG+NOC-1)
  200 CONTINUE
C  SCORE THE STUDENTS ITEM RESPONSES
    2 DO 600 NRESP=1,ITOT
      NOUTCO=IWORD(NRESP+IOUTBG-1)
      IF(IRESP(NRESP).EQ.IBLANK.AND.NOUTCO.NE.9) GO TO 24
      IF(IRESP(NRESP).NE.IWORD(NRESP+KEYBG-1)) GO TO 600
      MOUTCO=LPACK(NOUTCO)
      SCORES(NPOS,MOUTCO)=SCORES(NPOS,MOUTCO)+1
      SCORES(NPOS,10)    =SCORES(NPOS,10)    +1
  600 CONTINUE
      RETURN
   24 CALL ERR(828,24,ISIG(NPOS),0,0,0)
   25 LPROG=-99
      LUNIT=-99
   28 ISIG(NPOS)=0
```

```
 25 SCORES(NPOS,I)=0
    IREL=IREL-1
    RETURN
    END
    SUBROUTINE FIND(IER)
    COMMON /IBLK2/ JD(6),ID(6)
    COMMON /IBLK12/ITYPE
    COMMON /JBLK28/ JEBUF(43)
    COMMON /JBLK29/ ICODE,JCODE
    COMMON /KBLK5/ ICLASS,IDELE,IADD,IPAGE,ITEST,JTEST,
   *IREC,JOLD,JNEW,JRIDE
    COMMON /LBLK1/ KD(6)
    DIMENSION JBUFF(43),IBUFF(21)
    EQUIVALENCE (JEBUF(1),JBUFF(1))
    DO 90 I=1,6
    IF(I.EQ.4) GO TO                                    90
    IF(ID(I)-KD(I))                                     95,90,95
 90 CONTINUE
    RETURN
 95 IER=0
    GOTO 5000
 99 IF(JCODE.NE.0) GOTO                                 3000
100 IF(ITYPE)                                           101,1005,103
101 JTYPE=-ITYPE
    GOTO                                                (1001,1002,1003,1004,1005),
   *    JTYPE
1001 CALL NAMIN
    GOTO                                                100
1002 GOTO                                               1001
1003 LEN=21
    CALL INPUT(2,9,IBUFF,LEN)
    JOLD=JOLD+1
    GOTO                                                100
1004 GO TO                                              1003
1005 CALL LDRIN
    GO TO                                               5000
 103 GO TO                                              (2001,2002,2003,2004,2005),
   *    ITYPE
2001 CALL LDROUT
    GO TO                                               100
2002 GO TO                                              2001
2003 CALL NAMOUT
    GO TO                                               100
2004 LEN=21
    CALL OUTPUT(2,9,IBUFF,LEN)
    JNEW=JNEW+1
    GO TO                                               100
2005 GO TO                                              2004
3000 LEN=43
    CALL OUTPUT(2,4,JBUFF,LEN)
    JCODE=0
    JNEW=JNEW+1
    GO TO                                               100
5000 DO 5010 I=1,6                                      5010
    IF(I.EQ.4) GO TO                                    5020,5010,99
    IF(ID(I)-JD(I))
5010 CONTINUE
    CALL                                                LDROUT
```

```
5020 IER=1
     DO 5021 I=1,6
5021 KD(I)=ID(I)
     CALL ERR(&53,-4,0,0,0,0)
  53 RETURN
     END
     SUBROUTINE NTRAN(IUNIT,JACT,KLEN,LBUFF,NREC)
     COMMON K4
     DIMENSION LBUFF(KLEN)
     IF(JACT.EQ.1) GO TO 10
     IF(JACT.EQ.2) GO TO 20
     IF(JACT.EQ.6) GO TO 30
     IF(JACT.EQ.10) GO TO 40
     IF(JACT.EQ.22) GO TO 50
  10 WRITE(IUNIT'K4) LBUFF
     GO TO 50
  20 READ(IUNIT'K4) LBUFF
     GO TO 50
  30 K4=K4+NREC/3
     FIND(IUNIT'K4)
     GO TO 50
  40 K4=1
  50 RETURN
     END
     SUBROUTINE LDR
     COMMON /IBLK2/ JD(6),ID(6)
     COMMON /IBLK3/ JDNAME(6,5),JSITE
     COMMON/KBLK5/ ICLASS,IDEL,IADD,IPAGE,ITEST,JTEST,
    *              IREC,JOLD,JNEW,JRIDE
     DIMENSION IBUFF(24)
     ENTRY LDRIN
     IY=0
     LEN=24
     DO 200 IG=1,2
     CALL INPUT(2,IG,IBUFF,LEN)
     JOLD=JOLD+1
     IF(IG.EQ.1) JSITE=IBUFF(18)
     DO 201 I=1,6
 201 JD(I)=IBUFF(I+18)
     IZ=1
     DO 200 IX=1,3
     IY=IY+1
     DO 200 IV=1,5
     IZ=IZ+1
 200 JDNAME(IY,IV)=IBUFF(IZ)
     RETURN
     ENTRY LDROUT
     IY=0
     LEN=24
     DO 400 IG=1,2
     IBUFF(18)=JSITE
     DO 401 I=1,6
 401 IBUFF(I+18)=JD(I)
     IZ=1
     DO 410 IX=1,3
     IY=IY+1
     DO 410 IV=1,5
     IZ=IZ+1
```

```
  410 IBUFF(IZ)=JDNAME(IY,IV)
      IBUFF(1)=IG
      JNEW=JNEW+1
  400 CALL OUTPUT(2,IG,IBUFF,LEN)
      RETURN
      END
      SUBROUTINE NAME
      COMMON /IBLK2/ JD(6),ID(6)
      COMMON /IBLK22/ JHI
      COMMON /JBLK30/ IEBUF(11,60)
      COMMON/KBLK5/ ICLASS,IDEL,IADD,IPAGE,ITEST,JTEST,
     *              IREC,JOLD,JNEW ,JRIDE
C  THE FOLLOWING LIST CONTAINS THE STUDENT NAME RECORD FORMAT
C        FIELD NO   FIELD NAME                                  EDIT CODE  TOTAL COLS
C              01   RECORD TYPE                                    I1            1
C              BLANK SPACES                                       1X            2
C              02   STUDENT NAME LETTERS  1-4                     A4            6
C              03   STUDENT NAME LETTERS  5-8                     A4           10
C              04   STUDENT NAME LETTERS  9-12                    A4           14
C              05   STUDENT NAME LETTERS 13-16                    A4           18
C              06   STUDENT NAME LETTERS 17-20                    A4           22
C              BLANK SPACE                                        1X           23
C              07   LAST NAME RECORD INDICATOR (BLANK=NOT         A1           24
C                   LAST, ASTERISK=LAST NAME IN CLASS
C                   BLANK SPACES                                 35X           59
C               8   COMMAND CODE                                 I2            61
C                   BLANK SPACES                                 2X            63
C               9   STATUS ( BLANK=ACTIVE, 0=DELETED             A1            64
C                      1=NOT ASSIGNED)
C              10   GROUP NUMBER                                 I2            66
C              11   STUDENT NUMBER                               I2            80
      DIMENSION IBUFF(11,60)
      DATA ISTAR/'*'/,IBLANK/' '/
      EQUIVALENCE (IBUFF(1,1),IEBUF(1,1))
      ENTRY NAMIN
      LEN=11
      I=0
  100 I=I+1
C  MAXIMUM SIZE OF CLASS ROSTER IS 60
      IF(I.EQ.61) GO TO                          150
      CALL INPUT(2,3,IBUFF(1,I),LEN)
C  JOLD COUNTS OLD STUDENT DATA BASE RECORDS
      JOLD=JOLD+1
      IF(IBUFF(7,I).EQ.ISTAR) GO TO              200
      GO TO                                      100
  150 CALL ERR(&151,-5,0,0,0,0)
  151 STOP
  200 IBUFF(7,I)=IBLANK
      JHI=I
      RETURN
      ENTRY NAMOUT
      LEN=11
      I=0
  300 I=I+1
      IF(I.EQ.JHI+1) RETURN
      IF(I.EQ.JHI) IBUFF(7,I)=ISTAR
      CALL OUTPUT (2,3,IBUFF(1,I),LEN)
C  JNEW COUNTS NEW STUDENT DATA BASE RECORDS
```

```
      JNEW=JNEW+1
      GO TO                                        300
      END
      SUBROUTINE UPDATE
      INTEGER CRITAB,TOTCRI
      COMMON /IBLK2/
     1                JD(6),ID(6)
      COMMON /IBLK7/
     1                          NAXTAB(10),
     1              CRITAB(10),TOTCRI,ITEMS,NOCTOT,IPACK(10)
      COMMON /IBLK11/ IERR(2,100),MREL
      COMMON /IBLK20/ JYEAR,JMONTH,JDAY,JDENT,JTIME,JOVER
      COMMON /IBLK21/ IYEAR,IMONTH,IDAY
      COMMON /IBLK23/ ISWRL
      COMMON /JBLK3/
     1               IRESP(28),IUNI,IPROG,IPTR
      COMMON /JBLK4/ ISCORE(60,10)
      COMMON /JBLK28/ JEBUF(43)
      COMMON /JBLK29/ ICODE,JCODE
      COMMON /JBLK30/ IEBUF(11,60)
      COMMON /KBLK2/ ISIG(60),JSIG(60),KSIG(60),LSIG(60)
      COMMON /KBLK4/ IREL,JREL,KREL,LREL
      COMMON /KBLK5/ ICLASS,IDELE,IADD,IPAGE,ITEST,JTEST,IREC,
     * JOLD,JNEW,JRIDE
      DIMENSION       IBUFF(43),JBUFF(43)
      DATA IBLANK/' '/
      DATA IYES/'Y'/
      EQUIVALENCE    (JEBUF(1),JBUFF(1))
C  THE FOLLOWING LIST CONTAINS THE STUDENT SCORES RECORD FORMAT
C        FIELD NO   FIELD NAME                          EDIT CODE  TOTAL COLS
C              01   RECORD TYPE                           I1          1
C                   BLANK SPACE                          1X          2
C                   SCORE ON FIRST OUTCOME               I2          4
C                   ......
C                   BLANK SPACE                          1X         15
C                   TOTAL SCORE ON ALL OUTCOMES          I3         18
C    08             BLANK SPACE                          1X         19
C                   ITEM RESPONSES                      28A1        47
C    09-36          BLANK SPACE                          1X         48
C                   YEAR                                 I2         50
C    37             MONTH                                I2         52
C    38             DAY                                  I2         54
C    39             BLANK SPACES                         5X         59
C                   COMMAND CODE                         I2         61
C    40             BLANK SPACE                          1X         62
C                   UNIT NO.                             I2         64
C    41             GROUP NO.                            I2         66
C    42             BLANK SPACE                         12X         78
C                   STUDENT NO.                          I2         80
C    43
      LNO=ISIG(IPTR)
      IGRP=IEBUF(10,LNO)
      ICODE=IUNI*100 + LNO
  716 IF(ICODE-JCODE) 749,750,751
  749 ISIG(IPTR)=IGRP*100+ISIG(IPTR)
      J=0
      DO 730 K=2,7
      IF(K-1.GT.NOCTOT)GOTO                        731
      IBUFF(K)=ISCORE(IPTR,K-1)
```

```
      GOTO 730
  731 IBUFF(K)=0
  730 CONTINUE
      IBUFF(08)=ISCORE(IPTR,10)
      IBUFF(37)=IYEAR
      IBUFF(38)=IMONTH
      IBUFF(39)=IDAY
      IBUFF(40)=0
      IBUFF(41)=IUNI
      IBUFF(42)=IGRP
      IBUFF(43)=LNO
      DO 737 L=1,ITEMS
  737 IBUFF(08+L)=IRESP(L)
      LL=ITEMS+1
      DO 738 L=LL,28
  738 IBUFF(L+8)=IBLANK
      LEN=43
      IBUFF(1)=4
      CALL OUTPUT(2,4,IBUFF,LEN)
      JNEW=JNEW+1
      LREL=LREL+1
      LSIG(LREL)=LNO
      RETURN
  750 CONTINUE
  150 CALL ERR(8747,10,LNO,0,0,0)
      ISIG(IPTR)=0
      DO 25 I=1,10
   25 ISCORE(IPTR,I)=0
      IREL=IREL-1
      RETURN
  751 LEN=43
      IF(JCODE.EQ.0) GOTO 752
      CALL OUTPUT(2,4,JBUFF,LEN)
      JNEW=JNEW+1
      IF(IUNI.EQ.JBUFF(41)) LREL=LREL+1
      IF(IUNI.EQ.JBUFF(41)) LSIG(LREL)=JBUFF(43)
  752 LEN=43
      CALL INPUT(2,4,JBUFF,LEN)
      JOLD=JOLD+1
      JCODE=  JBUFF(41)*100+JBUFF(43)
      GOTO 716
      END
      SUBROUTINE STATUS
      COMMON /IBLK23/ ISWRL
      COMMON /JBLK7/ IPAGE,LINES
      COMMON /JBLK30/ IEBUF(11,60)
      COMMON /KBLK2/ ISIG(60),JSIG(60),KSIG(60),LSIG(60)
      COMMON /KBLK4/ IREL,JREL,KREL,LREL
 1121 FORMAT(12X,I2,4X,5A4)
      M=99
      IUNI=20
      IPAGE=0
      LINES=0
      IDIFF=1
      CALL PAGE(IUNI,M)
       F(JREL.EQ.0) GOTO
      CALL TOP(1,IUNI,M)
      DO 110 I=1,JREL
```

```
      JNO=JSIG(I)
      IF(LINES.GT.55)                                    GOTO 170
  172 WRITE(ISWRL,1121) JNO,(IEBUF(J,JNO),J=2,6)
      LINES=LINES+1
      GOTO 110
  170 CALL SPACE
      CALL PAGE(IUNI,M)
      CALL TOP(1,IUNI,M)
      GOTO 172
  110 CONTINUE
  119 IF(KREL.EQ.0) GOTO                                 121
      IF(LINES.GT.44)  GOTO 140
      CALL TOP(2,IUNI,M)
  142 DO 120 I=1,KREL
      KNO=KSIG(I)
      IF(LINES.GT.55) GOTO          150
  152 WRITE(ISWRL,1121) KNO,(IEBUF(J,KNO),J=2,6)
      LINES=LINES+1
      GOTO 120
  150 CALL SPACE
      CALL PAGE(IUNI,M)
      CALL TOP(2,IUNI,M)
      GOTO 152
  120 CONTINUE
  121 CALL                                               SPACE
      RETURN
  140 CALL SPACE
      CALL PAGE(IUNI,M)
      CALL TOP(2,IUNI,M)
      GO TO 142
      END
      SUBROUTINE TOP(ISW,IUNI,M)
 1000 FORMAT('==',1X,I2)
 1051 FORMAT(' 1.',
     *4X,'THE FOLLOWING PUPILS HAVE BEEN DELETED FROM THE CLASS RECORD:'
     *)
 1052 FORMAT(' 2.',
     *4X,'THE FOLLOWING PUPILS HAVE BEEN ADDED TO THE CLASS RECORD:')
      COMMON /JBLK7/ IPAGE,LINES
      COMMON /IBLK23/ ISWRL
      IDIFF=3
      WRITE(ISWRL,1000) IDIFF
      LINES=LINES+3
      IDIFF=1
      IF(ISW.EQ.1) WRITE(ISWRL,1051)
      IF(ISW.EQ.2) WRITE(ISWRL,1052)
      LINES=LINES+1
      WRITE(ISWRL,1000) IDIFF
      LINES=LINES+1
      RETURN
      END
      SUBROUTINE ERROR
      INTEGER SORT
      EXTERNAL COMP4
  999 FORMAT(
     *4X,'IF YOU NEED ASSISTANCE INTERPRETING ANY OF THE FOLLOWING    '/
     *4X,'MESSAGES OR WITH ANY PART OF SWRL''S IMS PROGRAM, CALL DR.'/
     *4X,'GORDON GIBBS, TELEPHONE 757-9141, EXTENSION 73.')
```

```
1000 FORMAT('**',1X,I2)
1001 FORMAT(' 1.',
     *2X,'REQUESTS TO DELETE RECORD FOR',I2,' PUPIL(S) CANNOT BE PRO- '/      A
     *4X,'CESSED SINCE THE PUPIL NUMBERS ARE NOT CODED.  PLEASE       '/      A
     *4X,'CORRECT AND RESUBMIT FORMS.                                 ')      A
1002 FORMAT(' 2.',
     *2X,'PUPIL NUMBERS GREATER THAN 60 WERE  CODED ON REQUESTS TO    '/      A
     *4X,'DELETE RECORDS FOR ',I2,' PUPIL(S).  PLEASE CORRECT AND     '/      A
     *4X,'RESUBMIT FORMS.                                             ')      A
1003 FORMAT(' 3.',
     *2X,'TWO OR MORE DELETE REQUESTS HAVE BEEN RECEIVED FOR EACH OF  '/      A
     *4X,'THE FOLLOWING PUPILS.  RECORDS FOR THESE PUPILS HAVE BEEN   '/      A
     *4X,'DELETED.  IF THE REDUNDANT REQUESTS PERTAIN TO OTHER PUPILS,'/      A
     *4X,'PLEASE CORRECT AND RESUBMIT FORMS.                          ')      A
1004 FORMAT(' 4.',
     *2X,'REQUESTS TO DELETE RECORDS FOR PUPILS WITH THE FOLLOWING    '/      A
     *4X,'PUPIL NUMBERS CANNOT BE PROCESSED.  PUPILS WITH THESE NUM-  '/      A
     *4X,'BERS WERE NOT INITIALLY ADDED TO CLASS RECORDS OR THEIR     '/      A
     *4X,'RECORDS HAVE PREVIOUSLY BEEN DELETED.  PLEASE CORRECT AND   '/      A
     *4X,'RESUBMIT FORMS.                                             ')      A
1006 FORMAT(' 6.',
     *2X,'ONE OR MORE CRITERION EXERCISE RECORD SHEET(S) DID NOT HAVE '/
     *4X,'THE LIST NUMBER CODED.  PLEASE CORRECT AND RESUBMIT ALL     '/
     *4X,'FORMS.                                                      ')
1009 FORMAT(' 9.',
     *2X,'MORE THAN ONE LIST NUMBER ',I1,' HAS BEEN SUBMITTED.  PLEASE'/      A
     *4X,'CORRECT THE LIST NUMBERS IN ERROR AND RESUBMIT ALL FORMS.   ')      A
1010 FORMAT('10.',
     *2X,'ITEM RESPONSES FOR THE FOLLOWING PUPILS WERE SUBMITTED AT AN'/
     *4X,'EARLIER DATE. IF YOU WISH TO HAVE THESE NEW SCORES REPORTED'/
     *4X,'PLEASE CONTACT SWRL.                                        ')
1011 FORMAT('11.',
     *2X,'THE FOLLOWING PUPIL NUMBERS APPEAR ON CRITERION EXERCISE    '/      A
     *4X,'RECORD SHEETS.  THESE NUMBERS HAVE NOT BEEN ASSIGNED TO     '/      A
     *4X,'PUPILS ON YOUR CLASS RECORD.  COMPLETE CLASS IDENTIFICATION '/      A
     *4X,'FORMS TO ADD EACH PUPIL AND RECORD ONLY THE ITEM RESPONSES  '/      A
     *4X,'OF THESE PUPILS ON NEW CRITERION EXERCISE RECORD SHEETS.    '/      A
     *4X,'RESUBMIT ALL CORRECTED FORMS.                               ')      A
1012 FORMAT('12.',
     *2X,'THE FOLLOWING PUPIL NUMBERS APPEAR ON CRITERION EXERCISE    '/04/15/7
     *4X,'RECORD SHEETS.  THESE NUMBERS DO NOT CORRESPOND TO "ACTIVE" '/04/15/7
     *4X,'PUPILS IN YOUR CLASS RECORD.  PLEASE RECORD THE ITEM        '/04/15/7
     *4X,'RESPONSES OF THESE PUPILS IN THE CORRECT POSITION ON NEW    '/04/15/7
     *4X,'CRITERION EXERCISE RECORD SHEETS AND RESUBMIT FORMS, OR     ')04/15/7
     *4X,'CONTACT SWRL FOR ASSISTANCE.
1013 FORMAT('13.',
     *2X,'LIST NUMBER ',I1,' HAS BEEN INCORRECTLY CODED SINCE PUPIL   '/
     *4X,'NUMBERS FOR THIS LIST HAVE NOT BEEN ASSIGNED TO PUPILS ON   '/
     *4X,'YOUR CLASS RECORD.  PLEASE CORRECT AND RESUBMIT THE FORM.   ')
1014 FORMAT('14.',
     *2X,'THE FOLLOWING PUPIL NUMBERS APPEAR ON CRITERION EXERCISES.  '/
     *4X,'THESE NUMBERS HAVE NOT BEEN ASSIGNED TO PUPILS ON YOUR CLASS'/
     *4X,'RECORD. IF THIS PUPIL IS IN YOUR CLASS COMPLETE CLASS       '/
     *4X,'IDENTIFICATION FORMS TO ADD EACH PUPIL AND SUBMIT THESE     '/
     *4X,'CLASS IDENTIFICATION FORMS AND THE CRITERION EXERCISES.     ')
1015 FORMAT('15.',
     *2X,'THE FOLLOWING PUPIL NUMBERS APPEAR ON CRITERION EXERCISES.  '/04/15/7
     *4X,'THESE NUMBERS DO NOT CORRESPOND TO "ACTIVE" PUPILS IN YOUR  '/04/15/7
```

```
     *4X,'CLASS RECORD.  PLEASE CORRECT THE NUMBERS AND RESUBMIT FORMS'/04/15/7
     *4X,'OR CONTACT SWRL FOR ASSISTANCE.                              ')04/15/7
 1016 FORMAT('16.',
     *2X,'CRITERION EXERCISE RECORD SHEETS FOR THE FOLLOWING PUPILS    '/
     *4X,'DID NOT HAVE THE "TESTED" FIELD MARKED. HOWEVER, PROCESSING '/
     *4X,'CONTINUED BECAUSE SOME RESPONSES WERE MARKED.  IF THIS IS   '/
     *4X,'INCORRECT,  PLEASE NOTIFY SWRL.                             ')
 1017 FORMAT('17.',
     *2X,'THE ''DELETE'' FIELD WAS MARKED ON CRITERION EXERCISES FOR  '/
     *4X,'THE FOLLOWING PUPILS.  HOWEVER,  THE EXERCISES WERE PROC-   '/
     *4X,'ESSED BECAUSE ITEM RESPONSES WERE MARKED.  IF THE PUPILS ARE'/
     *4X,'NO LONGER IN THE CLASS,  PLEASE SUBMIT ANOTHER DELETE       '/
     *4X,'REQUEST.                                                    ')
 1019 FORMAT('19.',                                                    CHG 11
     *2X,'BOTH THE ''TESTED'' AND THE ''DELETE'' FIELDS HAVE BEEN     '/
     *4X,'MARKED FOR THE FOLLOWING PUPILS.  IT WAS ASSUMED THAT THE   '/
     *4X,'''TESTED'' FIELD WAS DESIRED, AND THE TEST SHEET WAS        '/
     *4X,'SCORED WITHOUT DELETING THE PUPIL.  IF THIS IS INCORRECT    '/
     *4X,'PLEASE NOTIFY SWRL,                                         ')
 1020 FORMAT('20.',
     *2X,'CRITERION EXERCISES FOR THE PUPILS LISTED BELOW WERE NOT    '/
     *4X,'PROCESSED.  THE PUPIL NUMBER AND CAUSE ARE LISTED BELOW,    '/
     *4X,'PLEASE CORRECT AND RESUBMIT THESE FORMS.                    ')
 1021 FORMAT('   ',
     *9X,I2,' PAGES WITH BLANK PUPIL CODES                            ')
 1022 FORMAT('   ',
     *9X,I2,' PAGE 1''S FOR STUDENT NUMBER ',I2                        )
 1023 FORMAT('   ',
     *9X,I2,' PAGE 2''S FOR STUDENT NUMBER ',I2                        )
 1024 FORMAT('   ',
     *9X,'    UNMATCHED PAGE FOR PUPIL NUMBER ',I2                     )
 1025 FORMAT('   ',
     *9X,I2,' PUPIL NUMBER(S) GREATER THAN 60 ON PAGE NUMBER ',I2      )
 1026 FORMAT('26.',
     *2X,'ON ',I2,' PAGE(S) INVALID PROGRAM AND/OR UNIT NUMBERS HAVE  '/
     *4X,'BEEN ENCOUNTERED.  PLEASE CHECK PRE-CODED AREA AT TOP OF    '/
     *4X,'PAGE AND ERASE STRAY MARKS, PLEASE RESUBMIT PAGE.  IF THIS  '/
     *4X,'PROBLEM RECURS PLEASE CALL SWRL FOR ASSISTANCE.             ')
      COMMON /IBLK11/ IERR(2,100),MREL
      COMMON /IBLK22/ JHI
      COMMON /IBLK23/ ISWRL
      COMMON /JBLK3/ IRESP(28),IUNI,NPROG,NPOS
      COMMON /JBLK7/ IPAGE,LINES
      COMMON /JBLK30/ IEBUF(11,60)
      COMMON /KBLK2/ ISIG(60),JSIG(60),KSIG(60),LSIG(60)
      COMMON /KBLK4/ IREL,JREL,KREL,LREL
      DIMENSION KEY(100)
      DIMENSION IHEAD(100)
      DIMENSION ISTATU (60)
      DATA IZERO/'0'/,IBLANK/' '/
      ICOUNT=0
      LINES=0
      IDIFF=1
      I=SORT(KEY,IHEAD,MREL,COMP4)
C     LINE COUNTER, PROGRAM PAGES AT LINE 50 PRIOR TO PRINTING AN ERROR MESSAGE
C                   PROGRAM PAGES AT LINE 55 PRIOR TO PRINTING A STUDENT NAME
C    IERR(2,100) IS A LIST OF STUDENT NUMBERS TOGETHER WITH THEIR ERROR CODE
C    IERR(1,*) CONTAINS THE ERROR CODE NUMBER
```

```
C  IERR(2,*)CONTAINS THE STUDENT NUMBER
C   REDUNDANT ERROR MESSAGES FOR THE SAME STUDENT ARE SUPPRESSED
      M=98
C  IS SET TO 98 SO THAT THE CORRECT PAGE HEADINS WILL BE GENERATED
C BY THE PAGE SUBROUTINE
      GOTO                                          199
  106 IF(LINES.GT.45) GO TO 107                                          U
      GO TO 108                                                          U
  107 CALL SPACE
  199 CALL                                          PAGE(IUNI,M)
      WRITE(ISWRL,998)
      IDIFF=2
      WRITE(ISWRL,1000) IDIFF
      IDIFF=1
      LINES=LINES+5                                                      U
  108 ISUB=IERR(1,I)
C  PRINT ERROR MESSAGE                                                   U
      GO TO                                         (1,2,3,4,5,6,7,8,9,10,   U
     *                                              11,12,13,14,15,16,17,18,19,  UA
     *                                              20,21,22,23,24,25,26,27),    AU
     *                                              ISUB                         UU
    1 CALL LIZ1(KEY,I,ICOUNT,1)                                          U
      WRITE(ISWRL,1001) ICOUNT                                           U
      LINES=LINES+3                                                      U
      GO TO 400
    2 CALL LIZ1(KEY,I,ICOUNT,2)
      WRITE(ISWRL,1002) ICOUNT
      LINES=LINES+3                                                      U
      GO TO 400                                                          U
    3 WRITE(ISWRL,1003)
      WRITE(ISWRL,1000) IDIFF
      LINES=LINES+5
      CALL LIZ2( &3,KEY,I,ICOUNT,3)                                      U
      GO TO 400                                                          U
    4 WRITE(ISWRL,1004)                                                  U
      WRITE(ISWRL,1000) IDIFF
      LINES=LINES+6
      ICOUNT=-1
      CALL LIZ2( &4,KEY,I,ICOUNT,4)                                      U
      GO TO 400                                                          A
    5 CONTINUE                                                           U
    6 CALL LIZ1(KEY,I,ICOUNT,6)                                                04/15/72
      WRITE(ISWRL,1006)
      LINES=LINES+3                                                      U
      GO TO 400                                                          A
    7 CONTINUE                                                           A
    8 CONTINUE                                                           A
    9 CALL LIZ3(KEY,I,ICOUNT,9)                                          A
      WRITE(ISWRL,1009) IERR(2,I)
       LINES=LINES+2                                                     U
      GO TO 400
   10 WRITE(ISWRL,1010)                                                  U
      WRITE(ISWRL,1000) IDIFF
      LINES=LINES+4
      CALL LIZ2(&10,KEY,I,ICOUNT,10)                                     U
      GO TO 400                                                          U
   11 WRITE(ISWRL,1011)                                                  U
      WRITE(ISWRL,1000) IDIFF
```

```
      LINES=LINES+7
      ICOUNT=-1
      CALL LIZ2(&11,KEY,I,ICOUNT,11)
      GO TO 400
   12 WRITE(ISWRL,1012)
      WRITE(ISWRL,1000) IDIFF
      LINES=LINES+7
      ICOUNT=-1
      CALL LIZ2(&12,KEY,I,ICOUNT,12)
      GO TO 400
   13 CALL LIZ3(KEY,I,ICOUNT,13)
      ICOUNT=IERR(2,I)
      WRITE(ISWRL,1013)          ICOUNT
      LINES=LINES+3
      GO TO 400
   14 WRITE(ISWRL,1014)
      WRITE(ISWRL,1000) IDIFF
      LINES=LINES+6
      ICOUNT=-1
      CALL LIZ2(&14,KEY,I,ICOUNT,14)
      GO TO 400
   15 WRITE(ISWRL,1015)
      WRITE(ISWRL,1000) IDIFF
      LINES=LINES+5
      ICOUNT=-1
      CALL LIZ2(&15,KEY,I,ICOUNT,15)
      GO TO 400
   16 WRITE(ISWRL,1016)
      WRITE(ISWRL,1000) IDIFF
      LINES=LINES+5
      CALL LIZ2(&16,KEY,I,ICOUNT,16)
      GO TO 400
   17 WRITE(ISWRL,1017)
      WRITE(ISWRL,1000) IDIFF
      LINES=LINES+6
      CALL LIZ2(&17,KEY,I,ICOUNT,17)
      GO TO 400
   18 CONTINUE
   19 WRITE(ISWRL,1019)                                        CHG 1
      WRITE(ISWRL,1000) IDIFF                                  CHG 1
      LINES=LINES+6                                            CHG 1
      CALL LIZ2(&19,KEY,I,ICOUNT,19)                           CHG 1
      GOTO                                  400                CHG 1
   20 CALL LIZ1(KEY,I,ICOUNT,20)
      WRITE(ISWRL,1020)
      LINES=LINES+3
      GOTO 400
   21 CALL LIZ1(KEY,I,ICOUNT,21)
      WRITE(ISWRL,1021) ICOUNT
      LINES=LINES+1
      GOTO 400
   22 CALL LIZ3(KEY,I,ICOUNT,22)
      WRITE(ISWRL,1022) ICOUNT,IERR(2,I)
      LINES=LINES+1
      GOTO 400
   23 CALL LIZ3(KEY,I,ICOUNT,23)
      WRITE(ISWRL,1023) ICOUNT,IERR(2,I)
      LINES=LINES+1
```

```
      GOTO 400
   24 CALL LIZ3(KEY,I,ICOUNT,24)
      WRITE(ISWRL,1024) IERR(2,I)
      LINES=LINES+1
      GOTO 400
   25 CALL LIZ3(KEY,I,ICOUNT,25)
      WRITE(ISWRL,1025) ICOUNT,IERR(2,I)
      LINES=LINES+1
      GOTO 400
   26 CALL LIZ1(KEY,I,ICOUNT,26)
      WRITE(ISWRL,1026) ICOUNT
      LINES=LINES+4
      GOTO 400
   27 CONTINUE
  400 IDIFF=2
      ICOUNT=0
      WRITE(ISWRL,1000) IDIFF
      LINES=LINES+2
      IDIFF=1
      IF(I.EQ.0) GO TO 410
      I=KEY(I)
      IF(I.EQ.0) GO TO 410
      GOTO                                    106
  410 CALL                                    SPACE
      RETURN
      END
      SUBROUTINE ERR(*,LERR,LSTUD,LUNIT,LPAGE,LOK)
      COMMON /IBLK2/ JD(6),ID(6)
      COMMON /IBLK11/ IERR(2,100),MREL
 3002 FORMAT(' PAGE OR STUDENT NUMBER = ',I2///)
 4001 FORMAT(' SYS. ERR.- INVALID PROGRAM                    '///)          04/13/72
 4002 FORMAT(' SYS. ERR.- MISSING PAGES FOR STUDENT NUMBER= ',I2///)
 4003 FORMAT(' SYS. ERR.- MORE THAN 27 ITEMS ON PAGE OR TEST'///)
 4004 FORMAT(' SYS. ERR.- CLASS ID.= ',612,' IS NOT IN DATA BASE'///)
 4005 FORMAT(' SYS. ERR.- ATTEMPT TO ADD STUDENT NO. > 60 TO ROSTER'///)
 4006 FORMAT(' SYS. ERR.- MORE THAN 500 DATA RECORDS IN RUN'///)
 4007 FORMAT(' SYS. ERR.- NEW STUDENT NO. = ',I2,' HAS NO NAME'///)
 4008 FORMAT(' SYS. ERR.- STUDENT NO.= ',I2,' HAS UNMARKED DATE GRID'//)  CHG 12
 5001 FORMAT(' TEACHER ERROR MSG. NO.= ',I2)
      IF(LERR.LT.0) GOTO 500
      MREL=MREL+1
      IERR(1,MREL)=LERR
      IERR(2,MREL)=LSTUD
      IF(LERR.LT.20.OR.LERR.GT.25) GOTO 498
      MREL=MREL+1
      IERR(1,MREL)=20
      IERR(2,MREL)=0
  498 WRITE(6,5001) LERR
      WRITE(6,3002) LSTUD
      RETURN 1
  500 KERR=-LERR
      GO TO (8001,8002,8003,8004,8005,
     *     8006,8007,8008),KERR                                        CHG 12
 8001 WRITE(6,4001)
      RETURN 1
 8002 WRITE(6,4002)   LSTUD
      RETURN 1
 8003 WRITE(6,4003)
```

```
8004 WRITE(6,4004)   ID
     RETURN 1
8005 WRITE(6,4005)
     RETURN 1
     RETURN 1
8007 WRITE(6,4007) LSTUD
     RETURN 1                                                    CHG 12
8008 WRITE(6,4008) LSTUD                                         CHG 12
     RETURN 1
     END
     SUBROUTINE LIZ
     COMMON/IBLK11/IERR(2,100),MREL
     COMMON/IBLK23/ISWRL
     COMMON/JBLK3/IRESP(28),IUNI,NPROG,NPOS
     COMMON/JBLK7/IPAGE,LINES
     COMMON/JBLK30/IEBUF(11,60)
     DIMENSION KEY(100)
C
C  TALLY MATCHING CODES IN IERR ARRAY
C  ICOUNT=THE TALLY
C
     II=IPTR
10   IF(II.EQ.0) RETURN
     IF(IERR(1,II).EQ.ICODE) GO TO 20
     RETURN
20   ICOUNT=ICOUNT+1
     IPTR=II
     II=KEY(IPTR)
     GO TO 10
C
     ENTRY LIZ2(*,KEY,IPTR,ICOUNT,ICODE)
     JSWTCH=0
     IF(ICOUNT.EQ.-1) JSWTCH=1
     II=IPTR
C
C  WRITE OUT UNIQUE NAMES OF STUDENTS WITHIN THIS ICODE
C
22   IF(II.EQ.0) RETURN
25   IF(IERR(2,II).NE.ISTUD) GO TO 31
     IPTR=II
     II=KEY(IPTR)
     GO TO 22
30   ISWTCH=1
31   ISTUD=IERR(2,II)
     IF(LINES.GT.55) GOTO 33
     IF(JSWTCH.EQ.0)
    *WRITE(ISWRL,100) ISTUD,(IEBUF(I,ISTUD),I=2,6)
     IF(JSWTCH.EQ.1)
    *WRITE(ISWRL,101) ISTUD
101  FORMAT(10X,I2)
     LINES=LINES+1
     GO TO 25
 33  CALL SPACE
     CALL PAGE(IUNI,98)
C
     ENTRY LIZ3(KEY,IPTR,ICOUNT,ICODE)
     II=IPTR
C
```

```
C  TALLY LIKE STUDENTS WITHIN THIS CODE IN IERR ARRAY                    U
C                                                                       U
      ISTUD=IERR(2,II)                                                  U
      IF(II.EQ.0) RETURN                                                U
      IF(IERR(1,II).NE.ICODE)RETURN                                     U
      IF(IERR(2,II).NE.ISTUD)RETURN                                     U
      ICOUNT=ICOUNT+1                                                   U
      IPTR=II                                                           U
      II=KEY(IPTR)                                                      U
      GO TO 40                                                          U
      END                                                               U
      SUBROUTINE PAGE(IUNI,M)
      COMMON /IBLK3/ JDNAME(6,5),JSITE
      COMMON /IBLK20/ IYEAR,IMONTH,IDAY,JDENT,JTIME,JOVER
      COMMON /IBLK23/ ISWRL
      COMMON /IBLK24/ MSITE
      COMMON /JBLK3/ IRESP(28),IUNIT,JPROG,IPOS                         H4
      COMMON /JBLK7/ IPAGE,LINES
      DIMENSION JTAB(12)
      DATA      JTAB/' JAN',' FEB',' MAR',' APR',
     *              ' MAY','JUNE','JULY',' AUG',
     *              'SEPT',' OCT',' NOV',' DEC'/
      DATA IPD/'.'/,IBLANK/' '/
 3001 FORMAT(31X,'SCHOOL:',1X,4A4,A3)
 3002 FORMAT('  CLASS:',1X,4A4,A3,1X,'  SCHOOL:',1X,4A4,A3)
 4001 FORMAT('TEACHER:',1X,4A4,A3,1X,'DISTRICT:',1X,4A4,A3,4X,
     * A4,A1,I3,',',' 19',I2)
 5001 FORMAT(17X,'FYCSP REPORT PROCESSING CONDITIONS')
 5002 FORMAT(11X,'LMS HARPER & ROW 1 REPORT PROCESSING CONDITIONS')
 5003 FORMAT(11X,'LMS HARPER & ROW 2 REPORT PROCESSING CONDITIONS')
 5004 FORMAT( 7X,'LMS MACMILLAN BANK STREET REPORT PROCESSING CONDITIONS
     *')
 5005 FORMAT(11X,'LMS HARPER & ROW 3 REPORT PROCESSING CONDITIONS')     H4
 5007 FORMAT(20X)                                                       03/27/72
 6001 FORMAT(21X,'       CLASS RECORD REPORT')                          H4
 7001 FORMAT(11X,'THIS REPORT SHOWS THE PERCENTAGE PERFORMANCE'/
     *11X,'OF EACH PUPIL ON CRITERION EXERCISE UNIT',I3)
 7002 FORMAT(
     *6X,'THIS REPORT LISTS CONDITIONS WHICH MAY NEED TO BE ALTERED'/
     *6X,'IN ORDER TO CORRECTLY PROCESS CRITERION EXERCISE UNIT',I3)
 7003 FORMAT(15X,'THIS REPORT SHOWS PUPILS WHO HAVE BEEN'/
     *15X,'ADDED TO OR REMOVED FROM CLASS RECORDS')
 8001 FORMAT(19X,'FYCSP PUPIL PERFORMANCE REPORT')
 8002 FORMAT(12X,'LMS HARPER & ROW 1 PUPIL PERFORMANCE REPORT')
 8003 FORMAT(12X,'LMS HARPER & ROW 2 PUPIL PERFORMANCE REPORT')
 8004 FORMAT( 8X,'LMS MACMILLAN BANK STREET PUPIL PERFORMANCE REPORT')
 8005 FORMAT(12X,'LMS HARPER & ROW 3 PUPIL PERFORMANCE REPORT')         H4
 9010 FORMAT('==',1X,I2)
 9511 FORMAT('==',1X,'SITE',1X,I2)
      MONTH=IMONTH
      LINES=0
      IPAGE=0
      IF(MSITE.EQ.JSITE) GO TO 65
      WRITE(ISWRL,9511) JSITE
      MSITE=JSITE
   65 IDIFF=1
      WRITE(ISWRL,9010) IDIFF
      LINES=LINES + 1
```

```
      IPER=IPD
      IF(IMONTH.EQ.7) IPER=IBLANK
      WRITE(ISWRL,4001)
     *                    (JDNAME(3,J),J=1,5),(JDNAME(1,J),J=1,5),
     *JTAB(MONTH)    ,    IPER,IDAY,IYEAR
      LINES=LINES + 1
      IF(JDNAME(6,1).EQ.IBLANK)
     *WRITE(ISWRL,3001)
     *  (JDNAME(2,J),J=1,5)
      IF(JDNAME(6,1).NE.IBLANK)
     *WRITE(ISWRL,3002)
     *   (JDNAME(6,J),J=1,5),(JDNAME(2,J),J=1,5)
      LINES=LINES+1
      IDIFF=2
      WRITE(ISWRL,9010) IDIFF
      LINES=LINES + 2
      IF(M.EQ.99) GOTO 50
      IF (M.EQ.98) GOTO  2001
C  CRITERION EXERCISE                                                    H4
      GOTO (301,302,303,304,305),JPROG
  301 WRITE(ISWRL,8001)
      LINES=LINES + 1
      GOTO 350
  302 WRITE(ISWRL,8002)
      LINES=LINES + 1
      GOTO 350
  303 WRITE(ISWRL,8003)
      LINES=LINES + 1
      GOTO 350
  304 WRITE(ISWRL,8004)
      LINES=LINES + 1
      GOTO 350                                                          H
  305 WRITE(ISWRL,8005)                                                 H
      LINES=LINES + 1                                                   H
      GOTO 350
  350 IDIFF=2
      WRITE(ISWRL,9010) IDIFF
      LINES=LINES+2
      WRITE(ISWRL,7001) IUNI
      LINES=LINES + 2
      WRITE(ISWRL,9010) IDIFF
      LINES=LINES+2
      RETURN
C  STATUS CHANGES                                                        H
   50 CONTINUE
  101 WRITE(ISWRL,6001)
      LINES=LINES + 1
  150 IDIFF=2
      WRITE(ISWRL,9010)    IDIFF
      LINES=LINES+2
      WRITE(ISWRL,7003)
      LINES=LINES + 2
      WRITE(ISWRL,9010) IDIFF
      LINES=LINES+2
      RETURN
C  CONDITION REPORT                                                      03/27/7
 2001 IF(JPROG.LT.0 .OR.JPROG .GT.5) GO TO 207                          03/27/7
      GO TO (201,202,203,204,205),JPROG
```

```
201 WRITE(ISWRL,5001)
    LINES=LINES + 1
    GOTO 250
202 WRITE(ISWRL,5002)
    LINES=LINES + 1
    GOTO 250
203 WRITE(ISWRL,5003)
    LINES=LINES + 1
    GOTO 250
204 WRITE(ISWRL,5004)
    LINES=LINES + 1
    GOTO 250                                                        H4
205 WRITE(ISWRL,5005)                                               H4
    LINES=LINES + 1                                                 H4
    GOTO 250                                                    03/27/72
207 WRITE(ISWRL,5007)                                           03/27/72
    LINES=LINES + 1
250 IDIFF=2
    WRITE(ISWRL,9010) IDIFF
    LINES=LINES+2
    WRITE(ISWRL,7002) IUNI
    LINES=LINES + 2
    WRITE(ISWRL,9010) IDIFF
    LINES=LINES+2
    RETURN
    END
    SUBROUTINE REP
    INTEGER SORT
    INTEGER CRITAB,TOTCRI,PERC
    EXTERNAL COMP1
    COMMON /IBLK7/ NAXTAB(10),CRITAB(10),TOTCRI,ITEMS,
   1               NOUTS,IPACK(10)
    COMMON /IBLK23/ ISWRL
    COMMON /JBLK3/ IRESP(28),IUNI,NPROG,NPOS
    COMMON /JBLK7/ IPAGE,LINES
    COMMON /JBLK30/ IEBUF(11,60)
    COMMON /KBLK2/ ISIG(60),JSIG(60),KSIG(60),LSIG(60)
    COMMON /KBLK4/ IREL,JREL,KREL,MREL
    DIMENSION
   1KEY(60),ISTUD(10,2),IPERF(10,2),IACM(10,2),PERC(10),
   2IROMS(10),JROMS(10)
    DIMENSION IHEAD(60)
    DATA IROMS/ '   I',' II',' III',' IV',' V',' VI',
   *            ' VII','VIII',' IX',' X'/
8001 FORMAT('GROUP',I2,7X,'PUPILS',12X,'PART   ',2X,4(A4,2X))       H3
9001 FORMAT('GROUP',I2,7X,'PUPILS',12X,'OUTCOME',2X,4(A4,2X))
9004 FORMAT(9X,I2,3X,5A4,8X,4(I3,3X))                               YY
9006 FORMAT('GROUP',3X,'AVERAGE ',26X,4(I3,3X))                     YY
9009 FORMAT('CLASS   AVERAGE ',26X,4(I3,3X))
9010 FORMAT('==',1X,I2)
    DO 1005 J=1,NOUTS
    K=IPACK(J)
1005 JROMS(J)=IROMS(K)
    IPAGE=0
    IDIFF=1
    LINES=0
    M=-1
    DO 1015 J=1,10
```

```
      ISTUD(J,2)=0
      IPERF(J,2)=0
   15 IACM(J,2)=0
      I=SORT(KEY,IHEAD,IREL,COMP1)
      IPTR=I
      JNO=ISIG(IPTR)-100*(ISIG(IPTR)/100)
      CALL PAGE(IUNI,M)
 1052 DO 1025 J=1,10
      IPERF(J,1)=0
      IACM(J,1)=0
 1025 ISTUD(J,1)=0
      M=ISIG(IPTR)/100
      IF(LINES.GT.41) GOTO                              1091
      IF(NPROG.LT.5) GO TO 1092
      WRITE(ISWRL,8001)M,(JROMS(J),J=1,NOUTS)
      GO TO 1093
 1092 WRITE(ISWRL,9001) M,(JROMS(J),J=1,NOUTS)
 1093 WRITE(ISWRL,9010) IDIFF
      LINES=LINES+2
 1071 IF(LINES.GT.50) GOTO 1066
 1060 DO 1065 J=1,NOUTS
      ISTUD(J,1)=ISTUD(J,1)+1
      ISTUD(J,2)=ISTUD(J,2)+1
      IACM(J,1)=IACM(J,1)+ISCORE(IPTR,J)
      PERC(J)=
     *(ISCORE(IPTR,J)*100.)/(NAXTAB(J)*1.0) + .5
 1065 IACM(J,2)=IACM(J,2)+ISCORE(IPTR,J)
 1067 WRITE(ISWRL,9004) JNO,(IEBUF(J,JNO),J=2,6)
     * , (PERC(J),J=1,NOUTS)
      LINES=LINES+1
      IPTR=KEY(IPTR)
      IF(IPTR.EQ.0)GOTO 1070
      JNO=ISIG(IPTR)-100*(ISIG(IPTR)/100)
      IF(ISIG(IPTR)/100.NE.M) GOTO                      1070
      GOTO                                              1071
 1070 WRITE(ISWRL,9010) IDIFF
      LINES=LINES+1
      DO 1075 J=1,NOUTS
 1075 PERC(J)=
     *(((IACM(J,1)*1.0)/(ISTUD(J,1)*1.0))*100.)/
     *  (NAXTAB(J)*1.0)  + .5
      IF(M.EQ.0) GO TO 1080
      WRITE(ISWRL,9005)M,(PERC(J),J=1,NOUTS)
      LINES=LINES+1
      GO TO 1085
 1080 WRITE(ISWRL,9006)     (PERC(J),J=1,NOUTS)
      LINES=LINES + 1
 1085 IDIFF=2
      WRITE(ISWRL,9010) IDIFF
      LINES=LINES+2
      IDIFF=1
      IF(IPTR.NE.0) GO TO                               1052
      DO 1095 J=1,NOUTS
 1095 PERC(J)=
     *(((IACM(J,2)*1.0)/(ISTUD(J,2)*1.0))*100.)/
     *  (NAXTAB(J)*1.0)  + .5
      WRITE(ISWRL,9009) (PERC(J),J=1,NOUTS)
      LINES=LINES+1
```

H3
H3
K3

```
      CALL SPACE
      RETURN
 1091 CALL SPACE
      CALL PAGE(IUNI,M)
      GOTO                                          1092
 1066 CALL SPACE
      CALL PAGE(IUNI,M)
      IF(NPROG.LT.5) GO TO 1088                                        H3
      WRITE(ISWRL,8001)M,(JROMS(J),J=1,NOUTS)
      GO TO 1068                                                       H3
 1088 WRITE(ISWRL,9001)M, (JROMS(J),J=1,NOUTS)                         H3
 1068 WRITE(ISWRL,9010) IDIFF                                          H3
      LINES=LINES + 2
      GOTO                                          1060
      END
```

EXHIBIT 1

SUBROUTINE

COMMON BLOCK	Block data	Main PGM	IFACE	COMP1	COMP4	IOIBM	SCORE	FIND	NTRAN	LDR
IBLK2	X	X	X				X	X		X
IBLK3	X									X
IBLK7							X			
IBLK11	X	X			X		X			
IBLK12	X	X				X		X		
IBLK20		X	X							
IBLK21		X								
IBLK22	X	X								
IBLK23	X	X								
IBLK24	X	X								
JBLK3	X	X					X			
JBLK4		X					X			
JBLK7 JBLK25	X						X			
JBLK28								X		
JBLK29	X							X		
JBLK30		X								

SUBROUTINE

COMMON BLOCK

	Block data	Main PGM	IFACE	COMP1	COMP4	IOIBM	SCORE	FIND	NTRAN	LDR
KBLK1			X							
KBLK2		X		X			X			
KBLK4	X	X					X			
KBLK5	X	X						X		X
KBLK9			X							
LBLK1	X							X		
UNLABELED COMMON K4		X							X	

SUBROUTINE

COMMON BLOCK

	NAME	SPACE	UPDATE	COMP	STATUS	TOP	ERROR	ERR	LIZ	PAGE	REP
IBLK 2	X		X					X		X	
IBLK3										X	
IBLK7			X								X
IBLK11			X				X	X	X		
IBLK12											
IBLK20			X							X	
IBLK21			X								
IBLK22	X						X				
IBLK23		X	X		X	X	X		X	X	X
IBLK24										X	
JBLK3			X				X		X		X
JBLK4			X								X
JBLK7		X			X	X	X		X	X	X
JBLK25											
JBLK28			X								
JBLK29			X								
JBLK30	X		X		X		X		X		X

SUBROUTINE

COMMON BLOCK

	NAME	SPACE	UPDATE	COMP	STATUS	TOP	ERROR	ERR	LIZ	PAGE	REP
KBLK1				X							
KBLK2			X		X		X				X
KBLK4			X		X		X				X
KBLK5	X		X								
KBLK9				X							
LBLK1											
UNLABELED COMMON K4											

EXHIBIT 2

RECORD LAYOUT FORM

Type of Record DATA INPUT

Field No.	Field Name	Postion From	Thru	No. of Positions	Edit Code	Defined on Record Types	Comment
01	Port No.	1	2	2	I2	1,2, & 3	
02	Pupil No.	3	4	2	I2	3	
03	Program No.	5	6	2	I2	2 & 3	Value of 32 on record i
							ComSys 1 & 2 denotes ty
							2 record. In ComSys 3
							non-blank character in
							field 20 below indicate
							a type 2 record.
04	Criterion Exercise Unit No.	7	8	2	I2	3	
05	Page or List No.	9	9	1	I1	3	In ComSys 3 a list no.
							9 indicates that this
							field has been left bl
06	Delete Indicator	10	10	1	I1	3	
07	Tested Indicator	11	11	1	I1	3	Significant only on H&
							H&R2, and MBS.
08	Item Responses 1-4	12	15	4	A4	3	Last response is follo
							by an '*'.
09	Item Responses 5-8	16	19	4	A4	3	
10	Item Responses 9-12	20	23	4	A4	3	
11	Item Responses 13-16	24	27	4	A4	3	
12	Item Responses 17-20	28	31	4	A4	3	
13	Item Responses 21-24	32	35	4	A4	3	
14	Item Responses 25-28	36	39	4	A4	3	
-	BLANK				1X		

Comments:

EXHIBIT 2

RECORD LAYOUT FORM

Type of Record DATA INPUT

Field No.	Field Name	Postion From	Thru	No. of Positions	Edit Code	Defined on Record Types	Comment
15	District/School/	41	46	6	I6	1 & 2	
	Teacher ID numbers						
16	Unused	47	48	2	I2		
17	Grade/Class ID	49	52	4	I4	1 & 2	
	numbers						
18	Year/Month/Day	53	58	6	I6	1 & 2	
19	Group Number	59	60	2	I2	2	
20	Pupil name 1st four	61	64	4	A4	2	
	letters						
21	Pupil name 2nd four	65	68	4	A4	2	
	letters						
22	Pupil name 3rd four	69	72	4	A4	2	
	letters						
23	Pupil name 4th four	73	76	4	A4	2	
	letters						
24	Pupil name 5th four	77	80	4	A4	2	
	letters						

Comments:

EXHIBIT 3

RECORD LAYOUT FORM

Type of Record: SORTED DATA INPUT

Field No.	Field Name	Postion From	Thru	No. of Positions	Edit Code	Comment
01	Multiple page flag	1	2	2	I2	-8 indicates a multiple FYSCP c
						HR3 page -9 indicates multiple
02	Pupil code	3	4	2	I2	LMS list
03	Program number	5	6	2	I2	
04	Criterion exercise				I2	Set negative on delete and add
	unit number	7	8	2		pupil requests
05	Page or list number	9	9	1	I1	
06	Flag	10	10	1	I1	1 = delete
						2 = Processing Condition Repor
						message number 1019
						3 = Processing Conditions Repo
						message number 1021 or 2
						1025 otherwise the field i
						blank.
07	Tested Indicator	11	11	1	I1	
08-35	Item Responses 1-28	12	39	28	A1	Up to 27 item responses with t
						last response followed by a '
	Blank				1X	
36	District number	41	42	2	I2	
37	School number	43	44	2	I2	
38	Teacher number	45	46	2	I2	
39	Unused	47	48	2	I2	
40	Grade number	49	50	2	I2	
41	Class number	51	52	2	I2	
42	Year	53	54	2	I2	
43	Month	55	56	2	I2	

Comments:

EXHIBIT 3

RECORD LAYOUT FORM

Type of Record: SORTED DATA INPUT

Field No.	Field Name	Postion From	Thru	No. of Positions	Edit Code	Comment
44	Day	57	58	2	I2	
45	Group number	59	60	2	I2	
46	Pupil name letters	61	64	4	A4	
	1-4					
47	Pupil name letters	65	68	4	A4	
	5-8					
48	Pupil name letters	69	72	4	A4	
	9-16					
49	Pupil name letters	73	76	4	A4	
	13-16					
50	Pupil name letters	77	80	4	A4	
	17-20					

Comments:

EXHIBIT 4

RECORD LAYOUT FORM

Type of Record TEST DESCRIPTOR RECORD

Field No.	Field Name	Sample Record Contents	Intra-Record Word No.
01	Program number	01	1-1
02	Unit number	03	2-2
03	Field pointers*	09,29,49,53,57,58	3-8
04	Answer key	1,1,1,1,2,2,2,2,	9-28
		1,1,1,1,2,3,1,2,	
		3,1,2,3,2,1	
05	Outcome number	1,1,1,1,1,2,2,2,	29-48
		2,2,3,3,3,3,3,4,	
		4,4,4,4,	
06	No. items/Outcome	05,05,05,05	49-52
07	Criterion Level/	04,04,04,04	53-56
	Outcome		
08	Total No. of items	020	57-57
09	Criterion for Total	018	58-58
	(set to 0 if not		
	applicable		
10	Pad to the size of		59-84**
	the largest record		
	in the file		

Comments:

*Field pointers provide the relative start location of the variable length fields within the record. For example, the first field pointer, 9, indicates that the answer key field begins in record word 9.

**Currently 84 words

EXHIBIT 5

RECORD LAYOUT FORM

Type of Record: Record Type 1: Class Identification Record

Field No.	Field Name	Postion From	Thru	No. of Positions	Field Content (Contents of all numeric fields must be right justified)
1	Record Type	1	1	1	The required 1 in this field identifies this record as the Class Identification Record.
2	District Name	3	21	19	This alpha character name identifies the district to which this class data belongs
3	School Name	22	40	19	This alpha character name identified the school to which this class data belongs.
4	Teacher Name	41	59	19	This alpha character name identifies the teacher to which this class data belongs.
5	Site Identification Number	64	65	2	This two digit value identifies the site from which this class data was transmitted.
6	Glass Identification	67	78	12	Each successful digit pair within this concetrenation identifies respectively, the district (positions 67-68), school (positions 69-70), teacher (positions 71-72), program (positions 73-74), grade (positions 75-76), and class (positions 77-78) to which this class data belongs.

Comments:

EXHIBIT 6

RECORD LAYOUT FORM

Type of Record: Record Type 2: Continuation of the Class Identification Record

Field No.	Field Name	Postion From	Thru	No. of Positions	Field Content (Contents of all numeric fields must be right justified)
1	Record Type	1	1	1	The required 2 in this field identifies this record as the Continuation of the Class Identification Record.
2	Program Name(s)	3	21	19	This alpha character name(s) (usually abbreviate e.g. FYSCP, HR1) identifies the program(s) whic are being administered to the class.
3	Grade	22	40	19	This alpha character name (e.g. KINDERGARTEN) identifies the grade level to which this class data pertains.
4.	Class Name	41	59	19	This alpha character name (e.g. AM, PM) identif the class to which this class data pertains.
5	Class Identification Number	67	78	12	Each successive digit pair within this concaten tion identifies respectively, the district (positions 67-68), school (positions 68-70), teacher (positions 71-72), program (positions 73-74), grade (positions 75-76), and class (position 77-78).

Comments:

EXHIBIT 7

RECORD LAYOUT FORM

Type of Record: Record Type 3: Student Name Record

Field No.	Field Name	Postion From	Thru	No. of Positions	Field Content (Contents of all numeric fields must be right justified)
1	Record Type	1	1	1	The required 3 in this field identifies this record as the Pupil Name Record
2	Name	3	22	20	The name entered here will appear on all reports in which this field is reported.
3	Class Identification Number	67	78	12	Each successive digit pair within this concatenation identifies respectively, the district (positions 67-68), school (positions 69-70), teacher (positions 71-72), program (positions 73-74), grade (positions 75-76), and class (positions 77-78), to which this student's data belongs.
4	Pupil Number	79	80	2	This 2 digit value identifies the student to which this data pertains.

Comments:

EXHIBIT 8

RECORD LAYOUT FORM

Type of Record Record Type 4: Student Scores Record

Field No.	Field Description	Positions From	Positions Thru	No. of Positions	Field Content (Contents of all numeric fields must be right justified)
1	Record Type	1	1	1	The required 4 in this field identifies this record as the Pupil Scores Record.
2	First Outcome Score	3	4	2	Reflects the raw scores, 0, 1, 2, ...
3	Second Outcome Score	5	6	2	
4	Third Outcome Score	7	8	2	
5	Fourth Outcome Score	9	10	2	
6	Fifth Outcome Score	11	12	2	
7	Sixth Outcome Score	13	14	2	
8	Total Score	16	18	3	This 3 digit value indicates the maximum score attainable for the entire test.
9	Item Responses	20	47	28	1 alphanumeric character for each item
10	Date	49	54	6	Each digit pair within this concatenation identifies respectively the month (columns 56-57 day (columns 58-59), and year (columns 60-61) on which the unit test was administered.

Comments:

EXHIBIT 8

RECORD LAYOUT FORM

Type of Record Record Type 4: Student Scores Record

Field No.	Field Description	Positions From	Positions Thru	No. of Positions	Field Content (Contents of all numeric fields must be right justified)
11	Command Code	60	61	2	Code used by IMS Maintenance program to insert records. This is an A2 field
12	Unit Number	63	64	2	This 2 digit value identifies the unit number to which these test responses correspond.
13	Group Number	66	66	2	This 2 digit value identifies the group to which the student was assigned at the time of the test
14	Class Identification Number	67	78	12	Each successive digit pair within this concatenation identifies respectively the district (positions 67-68), school (positions 69-70), teacher (positions 71-72), program (positions 73-74), grade (positions 75-76), and class (positions 77-78).
15	Pupil Number	79	80	2	This 2 digit value identifies the student to which this data pertains.

Comments:

EXHIBIT 9

```
== SITE 99
==  1
TEACHER: MAUREEN GOOSE          DISTRICT: ROWLAND                    JUNE   2, 1972
                                  SCHOOL: HOLLINGWORTH
==  2
            LMS HARPER & ROW 2 PUPIL PERFORMANCE REPORT
==  2
           THIS REPORT SHOWS THE PERCENTAGE PERFORMANCE
           OF EACH PUPIL ON CRITERION EXERCISE UNIT 11
==  2
GROUP 0          PUPILS                 OUTCOME      I
==  1
          11   SHANABARGER, ERIC                    100
          12   BASICH, FRANK                         60
          13   BURRESS, DEWAIN                       90
          14   DAVIS, KEVIN                         100
          15   CRUZ, FRED DE LE                     100
          16   DENNIS, LAURA                        100
          17   GOMEZ, ANTHONY                        40
          18   HERNANDEZ, MICHAEL                    90
          19   HUGHES, KAREN                         90
          20   KNOERNSCHILD, BOB                     90
          21   SCHOFER, NICK                        100
          22   SHIELDS, ANDY                        100
          23   LA LONDE, MICHELLE                   100
==  1
GROUP    AVERAGE                                   89
==  2
GROUP 3          PUPILS                 OUTCOME      I
==  1
          24   NAPPER JACKIE                        100
==  1
GROUP 3 AVERAGE                                   100
==  2
CLASS    AVERAGE                                   90
== 23

==  1
TEACHER: MAUREEN GOOSE          DISTRICT: ROWLAND                    JUNE   2, 1972
                                  SCHOOL: HOLLINGWORTH
==  2
            LMS HARPER & ROW 2 PUPIL PERFORMANCE REPORT
==  2
           THIS REPORT SHOWS THE PERCENTAGE PERFORMANCE
           OF EACH PUPIL ON CRITERION EXERCISE UNIT 12
==  2
GROUP 0          PUPILS                 OUTCOME      I
==  1
           1   BAEZA, SUSAN                         100
           2   BACA, CATHERINE                      100
           3   FREATMAN, CRYSTAL                     80
           5   HART, JEAN                            90
           6   JUAREZ, JAMES                        100
           7   KEITH, WILLIAM                        90
           8   LIONTAS, ARTHUR                      100
           9   LOPEZ, PETER                         100
          10   MAIORANO, DEBBIE                     100
==  1
GROUP    AVERAGE                                   96
==  2
```

NOTE The == SITE record directs the 690 to transmit all following lines to the site number designated.

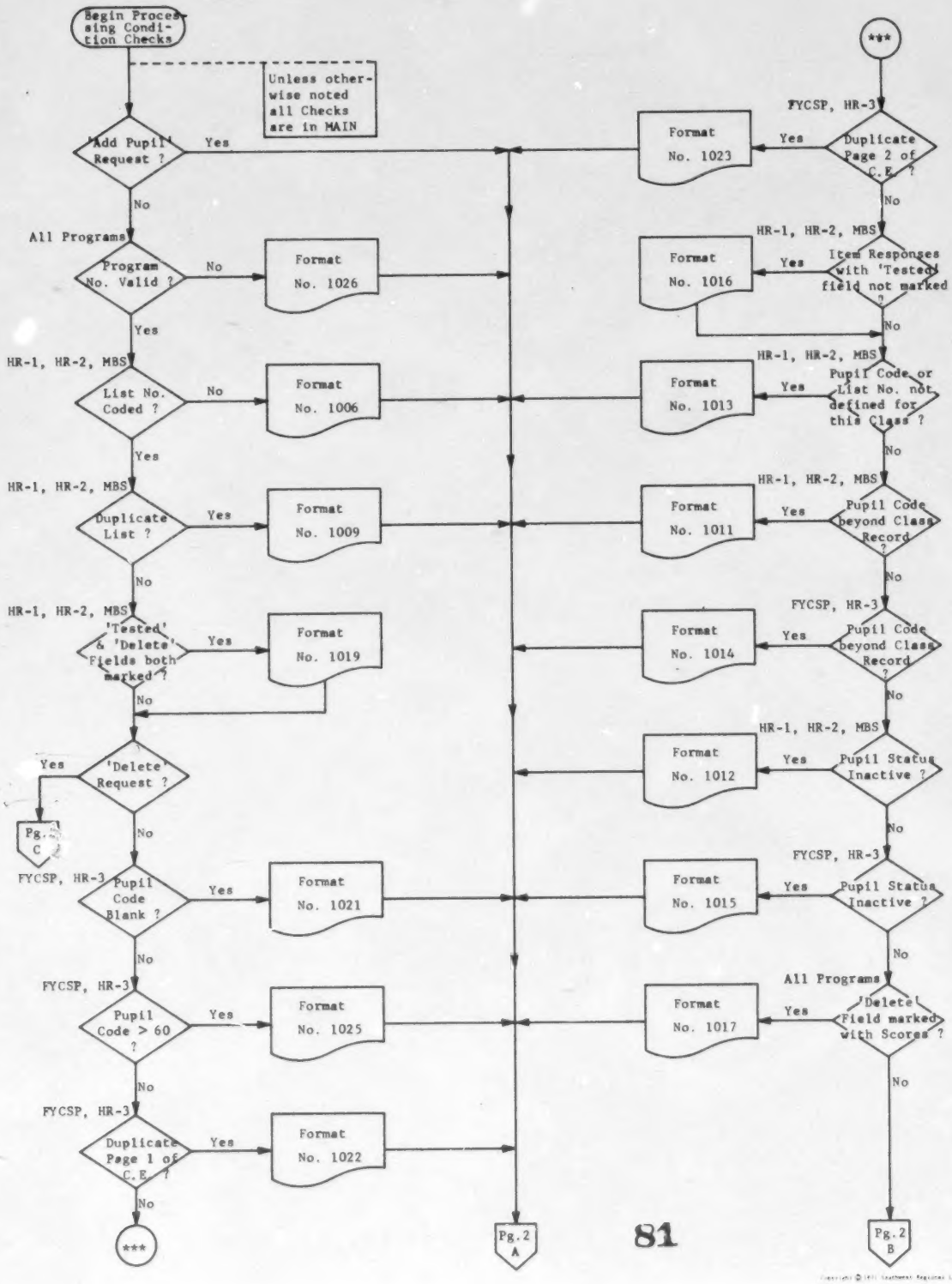
The == N where N is an integer direct the 690 to insert N blank lines into the report text on transmission to remote site.

```
1003 FORMAT(' 3.',
     *2X,'TWO OR MORE DELETE REQUESTS HAVE BEEN RECEIVED FOR EACH OF  '/
     *4X,'THE FOLLOWING PUPILS.  RECORDS FOR THESE PUPILS HAVE BEEN  '/
     *4X,'DELETED.  IF THE REDUNDANT REQUESTS PERTAIN TO OTHER PUPILS,'/
     *4X,'PLEASE CORRECT AND RESUBMIT FORMS.                         ')
1004 FORMAT(' 4.',
     *2X,'REQUESTS TO DELETE RECORDS FOR PUPILS WITH THE FOLLOWING    '/
     *4X,'PUPIL NUMBERS CANNOT BE PROCESSED.  PUPILS WITH THESE NUM-  '/
     *4X,'BERS WERE NOT INITIALLY ADDED TO CLASS RECORDS OR THEIR     '/
     *4X,'RECORDS HAVE PREVIOUSLY BEEN DELETED.  PLEASE CORRECT AND   '/
     *4X,'RESUBMIT FORMS.                                            ')
1006 FORMAT(' 6.',
     *2X,'ONE OR MORE CRITERION EXERCISE RECORD SHEET(S) DID NOT HAVE '/
     *4X,'THE LIST NUMBER CODED.  PLEASE CORRECT AND RESUBMIT ALL     '/
     *4X,'FORMS.                                                     ')
1009 FORMAT(' 9.',
     *2X,'MORE THAN ONE LIST NUMBER ',I1,' HAS BEEN SUBMITTED.  PLEASE'/
     *4X,'CORRECT THE LIST NUMBERS IN ERROR AND RESUBMIT ALL FORMS.   ')
1010 FORMAT('10.',
     *2X,'ITEM RESPONSES FOR THE FOLLOWING PUPILS WERE SUBMITTED AT AN'/
     *4X,'EARLIER DATE.  IF YOU WISH TO HAVE THESE NEW SCORES REPORTED'/
     *4X,'PLEASE CONTACT SWRL.                                        ')
1011 FORMAT('11.',
     *2X,'THE FOLLOWING PUPIL NUMBERS APPEAR ON CRITERION EXERCISE    '/
     *4X,'RECORD SHEETS.  THESE NUMBERS HAVE NOT BEEN ASSIGNED TO     '/
     *4X,'PUPILS ON YOUR CLASS RECORD.  COMPLETE CLASS IDENTIFICATION '/
     *4X,'FORMS TO ADD EACH PUPIL AND RECORD ONLY THE ITEM RESPONSES  '/
     *4X,'OF THESE PUPILS ON NEW CRITERION EXERCISE RECORD SHEETS.    '/
     *4X,'RESUBMIT ALL CORRECTED FORMS.                               ')
1012 FORMAT('12.',
     *2X,'THE FOLLOWING PUPIL NUMBERS APPEAR ON CRITERION EXERCISE    '/
     *4X,'RECORD SHEETS.  THESE NUMBERS DO NOT CORRESPOND TO "ACTIVE" '/
     *4X,'PUPILS IN YOUR CLASS RECORD.  PLEASE RECORD THE ITEM        '/
     *4X,'RESPONSES OF THESE PUPILS IN THE CORRECT POSITION ON NEW    '/
     *4X,'CRITERION EXERCISE RECORD SHEETS AND RESUBMIT FORMS, OR     '/
     *4X,'CONTACT SWRL FOR ASSISTANCE.                                ')
1013 FORMAT('13.',
     *2X,'LIST NUMBER ',I1,' HAS BEEN INCORRECTLY CODED SINCE PUPIL   '/
     *4X,'NUMBERS FOR THIS LIST HAVE NOT BEEN ASSIGNED TO PUPILS ON   '/
     *4X,'YOUR CLASS RECORD.  PLEASE CORRECT AND RESUBMIT THE FORM.   ')
1014 FORMAT('14.',
     *2X,'THE FOLLOWING PUPIL NUMBERS APPEAR ON CRITERION EXERCISES.  '/
     *4X,'THESE NUMBERS HAVE NOT BEEN ASSIGNED TO PUPILS ON YOUR CLASS'/
     *4X,'RECORD.  IF THIS PUPIL IS IN YOUR CLASS COMPLETE CLASS      '/
     *4X,'IDENTIFICATION FORMS TO ADD EACH PUPIL AND SUBMIT THESE     '/
     *4X,'CLASS IDENTIFICATION FORMS AND THE CRITERION EXERCISES.     ')
```

```
1015 FORMAT('15.',
     *2X,'THE FOLLOWING PUPIL NUMBERS APPEAR ON CRITERION EXERCISES.  '/
     *4X,'THESE NUMBERS DO NOT CORRESPOND TO "ACTIVE" PUPILS IN YOUR  '/
     *4X,'CLASS RECORD.  PLEASE CORRECT THE NUMBERS AND RESUBMIT FORMS'/
     *4X,'OR CONTACT SWRL FOR ASSISTANCE.                             ')
1016 FORMAT('16.',
     *2X,'CRITERION EXERCISE RECORD SHEETS FOR THE FOLLOWING PUPILS   '/
     *4X,'DID NOT HAVE THE "TESTED" FIELD MARKED. HOWEVER, PROCESSING '/
     *4X,'CONTINUED BECAUSE SOME RESPONSES WERE MARKED.  IF THIS IS   '/
     *4X,'INCORRECT,  PLEASE NOTIFY SWRL.                             ')
1017 FORMAT('17.',
     *2X,'THE ''DELETE'' FIELD WAS MARKED ON CRITERION EXERCISES FOR  '/
     *4X,'THE FOLLOWING PUPILS.  HOWEVER,  THE EXERCISES WERE PROC-   '/
     *4X,'ESSED BECAUSE ITEM RESPONSES WERE MARKED.  IF THE PUPILS ARE'/
     *4X,'NO LONGER IN THE CLASS,  PLEASE SUBMIT ANOTHER DELETE       '/
     *4X,'REQUEST.                                                   ')
1019 FORMAT('19.',
     *2X,'BOTH THE ''TESTED'' AND THE ''DELETE'' FIELDS HAVE BEEN     '/
     *4X,'MARKED FOR THE FOLLOWING PUPILS.  IT WAS ASSUMED THAT THE   '/
     *4X,'''TESTED'' FIELD WAS DESIRED, AND THE TEST SHEET WAS        '/
     *4X,'SCORED WITHOUT DELETING THE PUPIL.  IF THIS IS INCORRECT    '/
     *4X,'PLEASE NOTIFY SWRL.                                         ')
1020 FORMAT('20.',
     *2X,'CRITERION EXERCISES FOR THE PUPILS LISTED BELOW WERE NOT    '/
     *4X,'PROCESSED.  THE PUPIL NUMBER AND CAUSE ARE LISTED BELOW.    '/
     *4X,'PLEASE CORRECT AND RESUBMIT THESE FORMS.                    ')
1021 FORMAT('   ',
     *9X,I2,' PAGES WITH BLANK PUPIL CODES                            ')
1022 FORMAT('   ',
     *9X,I2,' PAGE 1''S FOR STUDENT NUMBER ',I2                       )
1023 FORMAT('   ',
     *9X,I2,' PAGE 2''S FOR STUDENT NUMBER ',I2                       )
1024 FORMAT('   ',
     *9X,'    UNMATCHED PAGE FOR PUPIL NUMBER ',I2                    )
1025 FORMAT('   ',
     *9X,I2,' PUPIL NUMBER(S) GREATER THAN 60 ON PAGE NUMBER ',I2     )
1026 FORMAT('26.',
     *2X,'ON ',I2,' PAGE(S) INVALID PROGRAM AND/OR UNIT NUMBERS HAVE  '/
     *4X,'BEEN ENCOUNTERED.  PLEASE CHECK PRE-CODED AREA AT TOP OF    '/
     *4X,'PAGE AND ERASE STRAY MARKS. PLEASE RESUBMIT PAGE.  IF THIS  '/
     *4X,'PROBLEM RECURS PLEASE CALL SWRL FOR ASSISTANCE.             ')
```

FLOWCHART LAYOUT FORM

PROGRAMMER: Howard Wolfe
PROGRAM ID: C I M S - Processing Condition Checks
DATE: April 27, 1972
PAGE 1 OF 2



SWICI

Flowchart Layout Form

Programmer: Howard Wolfe
Program ID: C I M S - Processing Condition Checks
Date: April 27, 1972
Page 2 of 2

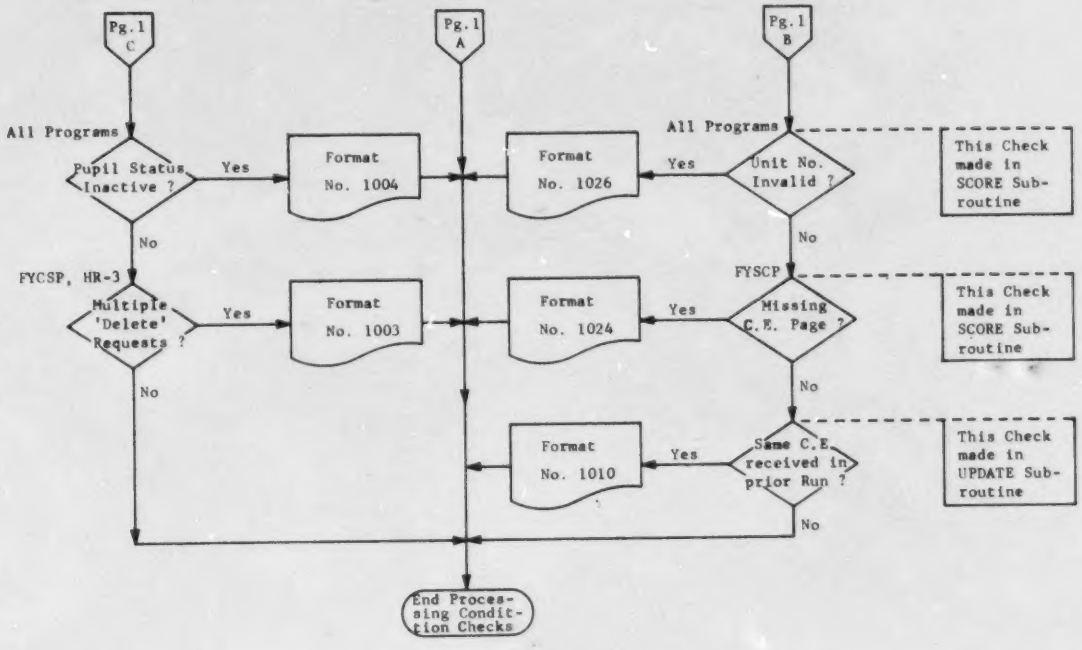


EXHIBIT 12

TEACHER: CYNTHIA KLINGER DISTRICT: TORRANCE SCHOOL: LEVY MAR, 3, 1972

LMS HARPER & ROW 1 REPORT PROCESSING CONDITIONS

THIS REPORT LISTS CONDITIONS WHICH MAY NEED TO BE ALTERED IN ORDER TO CORRECTLY PROCESS CRITERION EXERCISE UNIT 8

IF YOU NEED ASSISTANCE INTERPRETING ANY OF THE FOLLOWING MESSAGES OR WITH ANY PART OF SWRL'S IMS PROGRAM, CALL DR. BARBARA LASSER, TELEPHONE 776-3800, EXTENSION 50.

16. CRITERION EXERCISE RECORD SHEETS FOR THE FOLLOWING PUPILS DID NOT HAVE THE "TESTED" FIELD MARKED. HOWEVER, THE SHEETS WERE PROCESSED BECAUSE INCORRECT RESPONSES WERE MARKED. IF THIS IS INCORRECT, PLEASE NOTIFY SWRL.

6 MC DONALD, JENNIFER

NOTE The 690 control records (see EXHIBIT 9) have been removed.

EXHIBIT 13

TEACHER: ALMYRA DUKE

DISTRICT: ROWLAND
SCHOOL: HURLEY

MAR. 3, 1972

LMS HARPER & ROW 1 CLASS RECORD REPORT

THIS REPORT SHOWS PUPILS WHO HAVE BEEN
AD ED TO OR REMOVED FROM CLASS RECORDS

1. THE FOLLOWING PUPILS HAVE BEEN DELETED FROM THE CLASS RECORD:

25 TRACY, RALPH

2. THE FOLLOWING PUPILS HAVE BEEN ADDED TO THE CLASS RECORD:

27 GARCIA MARK
28 JOHNSON CYNTHIA

NOTE The 690 control records (see EXHIBIT 9) have been removed.

EXHIBIT 14

SYSTEM ERROR MESSAGES

1) 'SYS ERR - MORE THAN 27 ITEMS ON PAGE ON TEST'
This can only occur if there is a IMS software or transmission error. CIMS processing is halted.

2) 'SYS ERR - CLASS ID =_____ IS NOT IN DATA BASE'
This occurs if the Type 1, 2, and 3 records of the class being processed have not previously been inserted in the Pupil Data Base. This will also occur if the Class Id sheets have been recorded incorrectly or have had stray marks added to the ID fields. Class is skipped but processing continues.

3) 'SYS ERR - ATTEMPT TO ADD STUDENT NO. > 60 ROSTER'
Current version of CIMS can not have Class Record exceed 60. CIMS processing halts.

4) 'SYS ERR - MORE THAN 500 DATA RECORDS IN RUN'
Data Input File has exceeded 500 records. Records beyond 500 are dropped from the run.

5) 'SYS ERR - NEW STUDENT NO. = _____HAS UNMARKED DATE GRID'
Pupil Update form has been submitted but no pupil name has been coded. Processing of the update record is skipped.

6) 'SYS ERR - STUDENT NO. = _____HAS UNMARKED DATE GRID'
Processing continues and the date field is left blank. This field can be corrected in a Pupil Data Base Maintenance run.

EXHIBIT 15

SAMPLE RJS RUN SET UP

```
//AGEC 5HZ JOB '30,5000I,R,IL=SWRL,BIN=0738','WOLFE 757-9141 IMS'
//I S  EXEC LOADF,PG=250K,PG=200
//
//
//
```